

SSSSSSSSSSSS	MMM	MMM	GGGGGGGGGGGGG	RRRRRRRRRRRRR	TTTTTTTTTTTTTT	LLL	
SSSSSSSSSSSS	MMM	MMM	GGGGGGGGGGGGG	RRRRRRRRRRRRR	TTTTTTTTTTTTTT	LLL	
SSSSSSSSSSSS	MMM	MMM	GGGGGGGGGGGGG	RRRRRRRRRRRRR	TTTTTTTTTTTTTT	LLL	
SSS	MMMMMM	MMMMMM	GGG	RRR	TTT	LLL	
SSS	MMMMMM	MMMMMM	GGG	RRR	TTT	LLL	
SSS	MMMMMM	MMMMMM	GGG	RRR	TTT	LLL	
SSS	MM	MM	GGG	RRR	TTT	LLL	
SSS	MM	MM	GGG	RRR	TTT	LLL	
SSS	MM	MM	GGG	RRR	TTT	LLL	
SSS	MM	MM	GGG	RRR	TTT	LLL	
SSS	MM	MM	GGG	RRR	TTT	LLL	
SSSSSSSSSS	MM	MM	GGG	RRRRRRRRRRRR	TTT	LLL	
SSSSSSSSSS	MM	MM	GGG	RRRRRRRRRRRR	TTT	LLL	
SSSSSSSSSS	MM	MM	GGG	RRRRRRRRRRRR	TTT	LLL	
SSS	MM	MM	GGG	RRR	TTT	LLL	
SSS	MM	MM	GGG	RRR	TTT	LLL	
SSS	MM	MM	GGG	RRR	TTT	LLL	
SSS	MM	MM	GGG	RRR	TTT	LLL	
SSS	MM	MM	GGG	RRR	TTT	LLL	
SSS	MM	MM	GGG	RRR	TTT	LLL	
SSS	MM	MM	GGG	RRR	TTT	LLL	
SSS	MM	MM	GGG	RRR	TTT	LLL	
SSSSSSSSSS	MM	MM	GGGGGGGGGG	RRR	RRR	TTT	LLL
SSSSSSSSSS	MM	MM	GGGGGGGGGG	RRR	RRR	TTT	LLL
SSSSSSSSSS	MM	MM	GGGGGGGGGG	RRR	RRR	TTT	LLL

Val

001
001
001
001
001
001
001
001
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF

SSSSSSSS	MM	MM	GGGGGGGG	SSSSSSSS	TTTTTTTT	RRRRRRRR	TTTTTTTT	AAAAAA	BBBBBBBB
SSSSSSSS	MM	MM	GGGGGGGG	SSSSSSSS	TTTTTTTT	RRRRRRRR	TTTTTTTT	AAAAAA	BBBBBBBB
SS	MMMM	MMMM	GG	SS	TT	RR	RR	AA	BB
SS	MMMM	MMMM	GG	SS	TT	RR	RR	AA	BB
SS	MM MM	MM MM	GG	SS	TT	RR	RR	AA	BB
SS	MM MM	MM MM	GG	SS	TT	RR	RR	AA	BB
SSSSSS	MM	MM	GG	SSSSSS	TT	RRRRRRRR	TT	AA	BBBBBBBB
SSSSSS	MM	MM	GG	SSSSSS	TT	RRRRRRRR	TT	AA	BBBBBBBB
SS	MM	MM	GG GGGG	SS	TT	RR RR	TT	AAAAAAA	BB
SS	MM	MM	GG GGGG	SS	TT	RR RR	TT	AAAAAAA	BB
SS	MM	MM	GG	SS	TT	RR RR	TT	AA	BB
SS	MM	MM	GG	SS	TT	RR RR	TT	AA	BB
SSSSSSSS	MM	MM	GGGGGG	SSSSSSSS	TT	RR RR	TT	AA	BBBBBBBB
SSSSSSSS	MM	MM	GGGGGG	SSSSSSSS	TT	RR RR	TT	AA	BBBBBBBB

45

LL		SSSSSSSS
LL		SSSSSSSS
LL		SS
LL		SS
LL		SS
LL		SSSSSS
LL		SSSSSS
LL		SS
LL		SS
LL		SS
LLLLLLLL		SSSSSSSS
LLLLLLLL		SSSSSSSS

4F

4F

57

45

```

1 0001 0 MODULE SMG$STRING TABLES( XTITLE 'TPARSE tables for string capabilities'
2 0002 0 IDENT = '1-003'      ! File: SMGSTRTAB.B32 Edit: PLL1003
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 ****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 1 * ALL RIGHTS RESERVED.
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 1 * TRANSFERRED.
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 1 * CORPORATION.
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 1 *
26 0026 1 *
27 0027 1 ****
28 0028 1
29 0029 1
30 0030 1 ++
31 0031 1 FACILITY: Screen Management
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1 This module contains the LIB$TPARSE state tables used to parse
36 0036 1 string capabilities in an ascii TERMTABLE.TXT file.
37 0037 1
38 0038 1 ENVIRONMENT: User mode - AST reentrant
39 0039 1
40 0040 1 AUTHOR: P. Levesque CREATION DATE: 30-Jan-1984
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 1-001 - Original. PLL 30-Jan-1984
45 0045 1 -002 - Add SMG$K BEGIN NORMAL RENDITION. PLL 15-Mar-1984
46 0046 1 Enhance state tables to accept only !UL FAO directive.
47 0047 1 Allow comments on lines not terminated by a comma.
48 0048 1 -003 - Change implementation of control characters. PLL 5-Jul-1984
49 0049 1 --
50 0050 1

```

```

52 0051 1 XSBTTL 'Declarations'
53 0052 1 :
54 0053 1 : SWITCHES:
55 0054 1 :
56 0055 1 :
57 0056 1 :
58 0057 1 : LINKAGES:
59 0058 1 :
60 0059 1 : NONE
61 0060 1 :
62 0061 1 : TABLE OF CONTENTS:
63 0062 1 :
64 0063 1 :
65 0064 1 FORWARD ROUTINE
66 0065 1 CONVERT_CONTROL,
67 0066 1 CONVERT_ESCAPE,
68 0067 1 EXPRESSION_ERROR,
69 0068 1 END_ARITH_CAP,
70 0069 1 INIT_ARITH_CAP,
71 0070 1 INSERT_CARROT,
72 0071 1 INSERT_DOLLAR,
73 0072 1 INSERT_EXCLAMATION,
74 0073 1 INSERT_PARENTHESIS,
75 0074 1 INVALID_DIRECTIVE,
76 0075 1 NOT_STRING,
77 0076 1 STORE_OPERAND,
78 0077 1 STORE_SUBSTITUTION:
79 0078 1 :
80 0079 1 :
81 0080 1 : INCLUDE FILES:
82 0081 1 :
83 0082 1 :
84 0083 1 REQUIRE 'RTLIN:SMGPROLOG';
85 0161 1 :
86 0162 1 LIBRARY 'RTLML:SMGTPALIB';
87 0163 1 :
88 0164 1 LIBRARY 'RTLTPAMAC';
89 0165 1 :
90 0166 1 :
91 0167 1 : EQUATED SYMBOLS:
92 0168 1 :
93 0169 1 :
94 0170 1 LITERAL
95 0171 1 SINGLE_QUOTE = XX'27',
96 0172 1 DOUBLE_QUOTE = XX'22',
97 0173 1 LEFT_PAREN = XX'28',
98 0174 1 RIGHT_PAREN = XX'29';
99 0175 1 :
100 0176 1 :
101 0177 1 : FIELDS:
102 0178 1 :
103 0179 1 : NONE
104 0180 1 :
105 0181 1 : PSECTS:
106 0182 1 :
107 0183 1 :
108 0184 1 :

```

54
54
50
43

```

109 0185 1 : EXTERNAL REFERENCES:
110 0186 1 :
111 0187 1 : EXTERNAL ROUTINE
112 0188 1 OTSSCVT_T1_L,
113 0189 1 SMG$SBLEAKS_OFF,
114 0190 1 SMG$SBLEAKS_ON,
115 0191 1 SMG$SCOPY_CAP,
116 0192 1 SMG$SFLUSH_ARITHMETIC,
117 0193 1 SMG$SFLUSH_SAVED_BUFFER,
118 0194 1 SMG$SMISSING_END,
119 0195 1 SMG$SNEXT_RECORD,
120 0196 1 SMG$SAVE_TOKEN_STRING,
121 0197 1 SMG$STORE_CAP_MASK,
122 0198 1 SMG$SYNTAX_ERROR;
123 0199 1

124 0200 1 EXTERNAL
125 0201 1 SMGS_ERRAT_LIN,
126 0202 1 SMGS_INVDIR,
127 0203 1 SMGS_INVEXP,
128 0204 1 SMGS_MISTERNAME,
129 0205 1 SMGS_NOTSTRCAP,
130 0206 1 SMGS_SYNERR;
131 0207 1

132 0208 1 EXTERNAL
133 0209 1 SMG$SMASK_ADR,
134 0210 1 SMG$SNEXT_NEGATIVE_NUMBER,
135 0211 1 SMG$SCURRENT_LINE,
136 0212 1
137 0213 1 SMG$SDATA_OFFSET;
138 0214 1
139 0215 1 :
140 0216 1 : OWN STORAGE:
141 0217 1 :
142 0218 1 : NONE

```

```
144      0219 1 %SBTTL 'SMG$STRING_TABLES - TPARSE tables for string capabilities'
145      0220 1 ++
146      0221 1 FUNCTIONAL DESCRIPTION:
147      0222 1
148      0223 1 The following are the state tables used to parse string
149      0224 1 capabilities in a terminal definition.
150      0225 1
151      0226 1 --
152      0227 1
153      0228 1 $INIT_STATE (SMG$SA_STRING_STATES, SMG$SA_STRING_KEYWDS);
154      0229 1 ! set up state tables, key words
155      0230 1
156      0231 1 ++
157      0232 1 ! Begin scanning loop. Look for the start of a capability.
158      0233 1 ! Skip over blanks and comments.
159      0234 1 !-
160      0235 1
161      P 0236 1 $STATE (BEGIN_SCAN,
162      P 0237 1     ((END_OF [INE], BEGIN_SCAN, SMG$NEXT_RECORD),
163      P 0238 1     ('!', BEGIN_SCAN, SMG$NEXT_RECORD),
164      P 0239 1     ((CAPABILITY), BEGIN_SCAN, SMG$BLANKS_OFF),
165      P 0240 1     (TPAS_LAMBDA, TPAS_EXIT)
166      P 0241 1     );
167      P 0242 1
168      P 0243 1 ++
169      P 0244 1 ! This state indicates the end of a line. A comment also signals the
170      P 0245 1 ! end of a line.
171      P 0246 1 !-
172      P 0247 1
173      P 0248 1 $STATE (END_OF_LINE,
174      P 0249 1     (TPAS_EOS, TPAS_EXIT),
175      P 0250 1     ('!', TPAS_EXIT),
176      P 0251 1     (TPAS_LAMBDA, TPAS_FAIL)
177      P 0252 1     );
178      P 0253 1
179      P 0254 1 ++
180      P 0255 1 ! Find the capability name and determine if it's one that we expect.
181      P 0256 1 ! The string up to the '=' sign should be the capability name.
182      P 0257 1 !-
183      P 0258 1
184      P 0259 1 $STATE (CAPABILITY,
185      P 0260 1     ((STRING_NAME), EQUALS_STRING, SMG$BLANKS_OFF),
186      P 0261 1     ('END', TPAS_FAIL),
187      P 0262 1     ('BOOLEAN', TPAS_FAIL),
188      P 0263 1     ('NUMERIC', TPAS_FAIL),
189      P 0264 1     ('STRING', BEGIN_SCAN),
190      P 0265 1     ('REQUIRE', TPAS_FAIL, SMG$MISSING_END),
191      P 0266 1     ('NAME', TPAS_FAIL, SMG$MISSING_END),
192      P 0267 1     (TPAS_SYMBOL, NOT_STRING)
193      P 0268 1     );
194      P 0269 1
195      P 0270 1 ++
196      P 0271 1 ! Determine if string capability name is valid.
197      P 0272 1 !-
198      P 0273 1
199      P 0274 1 $STATE (STRING_NAME,
200      P 0275 1     ('BEGIN_ALTERNATE_CHAR', TPAS_EXIT, , SMG$K_BEGIN_ALTERNATE_CHAR, SMG$MASK_ADR),
```

```

: 201 P 0276 1 ('BEGIN_BLINK', TPAS_EXIT, . SMG$K BEGIN_BLINK, SMG$MASK_ADR),
: 202 P 0277 1 ('BEGIN_BOLD', TPAS_EXIT, . SMG$K BEGIN_BOLD, SMG$MASK_ADR),
: 203 P 0278 1 ('BEGIN_DELETE_MODE', TPAS_EXIT, . SMG$K BEGIN_DELETE_MODE, SMG$MASK_ADR),
: 204 P 0279 1 ('BEGIN_INSERT_MODE', TPAS_EXIT, . SMG$K BEGIN_INSERT_MODE, SMG$MASK_ADR),
: 205 P 0280 1 ('BEGIN_LINE_DRAWING_CHAR', TPAS_EXIT, . SMG$K BEGIN_LINE_DRAWING_CHAR, SMG$MASK_ADR),
: 206 P 0281 1 ('BEGIN_NORMAL_RENDERING', TPAS_EXIT, . SMG$K BEGIN_NORMAL_RENDERING, SMG$MASK_ADR),
: 207 P 0282 1 ('BEGIN_REVERSE', TPAS_EXIT, . SMG$K BEGIN_REVERSE, SMG$MASK_ADR),
: 208 P 0283 1 ('BEGIN_UNDERSCORE', TPAS_EXIT, . SMG$K BEGIN_UNDERSCORE, SMG$MASK_ADR),
: 209 P 0284 1 ('BOTTOM_T_CHAR', TPAS_EXIT, . SMG$K BOTTOM_T_CHAR, SMG$MASK_ADR),
: 210 P 0285 1 ('CLEAR_TAB', TPAS_EXIT, . SMG$K CLEAR_TAB, SMG$MASK_ADR),
: 211 P 0286 1 ('CR_GRAPHIC', TPAS_EXIT, . SMG$K CR_GRAPHIC, SMG$MASK_ADR),
: 212 P 0287 1 ('CROSS_CHAR', TPAS_EXIT, . SMG$K CROSS_CHAR, SMG$MASK_ADR),
: 213 P 0288 1 ('CURSOR_DOWN', TPAS_EXIT, . SMG$K CURSOR_DOWN, SMG$MASK_ADR),
: 214 P 0289 1 ('CURSOR_LEFT', TPAS_EXIT, . SMG$K CURSOR_LEFT, SMG$MASK_ADR),
: 215 P 0290 1 ('CURSOR_NEXT_LINE', TPAS_EXIT, . SMG$K CURSOR_NEXT_LINE, SMG$MASK_ADR),
: 216 P 0291 1 ('CURSOR_POSITION_REPORT', TPAS_EXIT, . SMG$K CURSOR_POSITION_REPORT, SMG$MASK_ADR),
: 217 P 0292 1 ('CURSOR_PRECEDING_LINE', TPAS_EXIT, . SMG$K CURSOR_PRECEDING_LINE, SMG$MASK_ADR),
: 218 P 0293 1 ('CURSOR_RIGHT', TPAS_EXIT, . SMG$K CURSOR_RIGHT, SMG$MASK_ADR),
: 219 P 0294 1 ('CURSOR_UP', TPAS_EXIT, . SMG$K CURSOR_UP, SMG$MASK_ADR),
: 220 P 0295 1 ('DARK_SCREEN', TPAS_EXIT, . SMG$K DARK_SCREEN, SMG$MASK_ADR),
: 221 P 0296 1 ('DELETE_CHAR', TPAS_EXIT, . SMG$K DELETE_CHAR, SMG$MASK_ADR),
: 222 P 0297 1 ('DELETE_LINE', TPAS_EXIT, . SMG$K DELETE_LINE, SMG$MASK_ADR),
: 223 P 0298 1 ('DEVICE_ATTRIBUTES', TPAS_EXIT, . SMG$K DEVICE_ATTRIBUTES, SMG$MASK_ADR),
: 224 P 0299 1 ('DOUBLE_HIGH_BOTTOM', TPAS_EXIT, . SMG$K DOUBLE_HIGH_BOTTOM, SMG$MASK_ADR),
: 225 P 0300 1 ('DOUBLE_HIGH_TOP', TPAS_EXIT, . SMG$K DOUBLE_HIGH_TOP, SMG$MASK_ADR),
: 226 P 0301 1 ('DOUBLE_WIDE', TPAS_EXIT, . SMG$K DOUBLE_WIDE, SMG$MASK_ADR),
: 227 P 0302 1 ('DUPLICATE', TPAS_EXIT, . SMG$K DUPLICATE, SMG$MASK_ADR),
: 228 P 0303 1 ('END_ALTERNATE_CHAR', TPAS_EXIT, . SMG$K END_ALTERNATE_CHAR, SMG$MASK_ADR),
: 229 P 0304 1 ('END_BLINK', TPAS_EXIT, . SMG$K END_BLINK, SMG$MASK_ADR),
: 230 P 0305 1 ('END_BOLD', TPAS_EXIT, . SMG$K END_BOLD, SMG$MASK_ADR),
: 231 P 0306 1 ('END_DELETE_MODE', TPAS_EXIT, . SMG$K END_DELETE_MODE, SMG$MASK_ADR),
: 232 P 0307 1 ('END_INSERT_MODE', TPAS_EXIT, . SMG$K END_INSERT_MODE, SMG$MASK_ADR),
: 233 P 0308 1 ('END_LINE_DRAWING_CHAR', TPAS_EXIT, . SMG$K END_LINE_DRAWING_CHAR, SMG$MASK_ADR),
: 234 P 0309 1 ('END_REVERSE', TPAS_EXIT, . SMG$K END_REVERSE, SMG$MASK_ADR),
: 235 P 0310 1 ('END_UNDERSCORE', TPAS_EXIT, . SMG$K END_UNDERSCORE, SMG$MASK_ADR),
: 236 P 0311 1 ('ERASE_TO_END_DISPLAY', TPAS_EXIT, . SMG$K ERASE_TO_END_DISPLAY, SMG$MASK_ADR),
: 237 P 0312 1 ('ERASE_TO_END_LINE', TPAS_EXIT, . SMG$K ERASE_TO_END_LINE, SMG$MASK_ADR),
: 238 P 0313 1 ('ERASE_WHOLE_DISPLAY', TPAS_EXIT, . SMG$K ERASE_WHOLE_DISPLAY, SMG$MASK_ADR),
: 239 P 0314 1 ('ERASE_WHOLE_LINE', TPAS_EXIT, . SMG$K ERASE_WHOLE_LINE, SMG$MASK_ADR),
: 240 P 0315 1 ('FF_GRAPHIC', TPAS_EXIT, . SMG$K FF_GRAPHIC, SMG$MASK_ADR),
: 241 P 0316 1 ('HOME', TPAS_EXIT, . SMG$K HOME, SMG$MASK_ADR),
: 242 P 0317 1 ('HORIZONTAL_BAR', TPAS_EXIT, . SMG$K HORIZONTAL_BAR, SMG$MASK_ADR),
: 243 P 0318 1 ('HT_GRAPHIC', TPAS_EXIT, . SMG$K HT_GRAPHIC, SMG$MASK_ADR),
: 244 P 0319 1 ('INDEX', TPAS_EXIT, . SMG$K INDEX, SMG$MASK_ADR),
: 245 P 0320 1 ('INIT_STRING', TPAS_EXIT, . SMG$K INIT_STRING, SMG$MASK_ADR),
: 246 P 0321 1 ('INSERT_CHAR', TPAS_EXIT, . SMG$K INSERT_CHAR, SMG$MASK_ADR),
: 247 P 0322 1 ('INSERT_LINE', TPAS_EXIT, . SMG$K INSERT_LINE, SMG$MASK_ADR),
: 248 P 0323 1 ('INSERT_PAD', TPAS_EXIT, . SMG$K INSERT_PAD, SMG$MASK_ADR),
: 249 P 0324 1 ('KEY_0', TPAS_EXIT, . SMG$K KEY_0, SMG$MASK_ADR),
: 250 P 0325 1 ('KEY_1', TPAS_EXIT, . SMG$K KEY_1, SMG$MASK_ADR),
: 251 P 0326 1 ('KEY_2', TPAS_EXIT, . SMG$K KEY_2, SMG$MASK_ADR),
: 252 P 0327 1 ('KEY_3', TPAS_EXIT, . SMG$K KEY_3, SMG$MASK_ADR),
: 253 P 0328 1 ('KEY_4', TPAS_EXIT, . SMG$K KEY_4, SMG$MASK_ADR),
: 254 P 0329 1 ('KEY_5', TPAS_EXIT, . SMG$K KEY_5, SMG$MASK_ADR),
: 255 P 0330 1 ('KEY_6', TPAS_EXIT, . SMG$K KEY_6, SMG$MASK_ADR),
: 256 P 0331 1 ('KEY_7', TPAS_EXIT, . SMG$K KEY_7, SMG$MASK_ADR),
: 257 P 0332 1 ('KEY_8', TPAS_EXIT, . SMG$K KEY_8, SMG$MASK_ADR),

```

```

: 258 P 0333 1 ('KEY_9', TPAS_EXIT, SMG$K_KEY_9, SMG$SMASK_ADR),
: 259 P 0334 1 ('KEY_BACKSPACE', TPAS_EXIT, SMG$K_KEY_BACKSPACE, SMG$SMASK_ADR),
: 260 P 0335 1 ('KEY_COMM', TPAS_EXIT, SMG$K_KEY_COMM, SMG$SMASK_ADR),
: 261 P 0336 1 ('KEY_DOWN_ARROW', TPAS_EXIT, SMG$K_KEY_DOWN_ARROW, SMG$SMASK_ADR),
: 262 P 0337 1 ('KEY_E1', TPAS_EXIT, SMG$K_KEY_E1, SMG$SMASK_ADR),
: 263 P 0338 1 ('KEY_E2', TPAS_EXIT, SMG$K_KEY_E2, SMG$SMASK_ADR),
: 264 P 0339 1 ('KEY_E3', TPAS_EXIT, SMG$K_KEY_E3, SMG$SMASK_ADR),
: 265 P 0340 1 ('KEY_E4', TPAS_EXIT, SMG$K_KEY_E4, SMG$SMASK_ADR),
: 266 P 0341 1 ('KEY_E5', TPAS_EXIT, SMG$K_KEY_E5, SMG$SMASK_ADR),
: 267 P 0342 1 ('KEY_E6', TPAS_EXIT, SMG$K_KEY_E6, SMG$SMASK_ADR),
: 268 P 0343 1 ('KEY_ENTER', TPAS_EXIT, SMG$K_KEY_ENTER, SMG$SMASK_ADR),
: 269 P 0344 1 ('KEY_F1', TPAS_EXIT, SMG$K_KEY_F1, SMG$SMASK_ADR),
: 270 P 0345 1 ('KEY_F2', TPAS_EXIT, SMG$K_KEY_F2, SMG$SMASK_ADR),
: 271 P 0346 1 ('KEY_F3', TPAS_EXIT, SMG$K_KEY_F3, SMG$SMASK_ADR),
: 272 P 0347 1 ('KEY_F4', TPAS_EXIT, SMG$K_KEY_F4, SMG$SMASK_ADR),
: 273 P 0348 1 ('KEY_F5', TPAS_EXIT, SMG$K_KEY_F5, SMG$SMASK_ADR),
: 274 P 0349 1 ('KEY_F6', TPAS_EXIT, SMG$K_KEY_F6, SMG$SMASK_ADR),
: 275 P 0350 1 ('KEY_F7', TPAS_EXIT, SMG$K_KEY_F7, SMG$SMASK_ADR),
: 276 P 0351 1 ('KEY_F8', TPAS_EXIT, SMG$K_KEY_F8, SMG$SMASK_ADR),
: 277 P 0352 1 ('KEY_F9', TPAS_EXIT, SMG$K_KEY_F9, SMG$SMASK_ADR),
: 278 P 0353 1 ('KEY_F10', TPAS_EXIT, SMG$K_KEY_F10, SMG$SMASK_ADR),
: 279 P 0354 1 ('KEY_F11', TPAS_EXIT, SMG$K_KEY_F11, SMG$SMASK_ADR),
: 280 P 0355 1 ('KEY_F12', TPAS_EXIT, SMG$K_KEY_F12, SMG$SMASK_ADR),
: 281 P 0356 1 ('KEY_F13', TPAS_EXIT, SMG$K_KEY_F13, SMG$SMASK_ADR),
: 282 P 0357 1 ('KEY_F14', TPAS_EXIT, SMG$K_KEY_F14, SMG$SMASK_ADR),
: 283 P 0358 1 ('KEY_F15', TPAS_EXIT, SMG$K_KEY_F15, SMG$SMASK_ADR),
: 284 P 0359 1 ('KEY_F16', TPAS_EXIT, SMG$K_KEY_F16, SMG$SMASK_ADR),
: 285 P 0360 1 ('KEY_F17', TPAS_EXIT, SMG$K_KEY_F17, SMG$SMASK_ADR),
: 286 P 0361 1 ('KEY_F18', TPAS_EXIT, SMG$K_KEY_F18, SMG$SMASK_ADR),
: 287 P 0362 1 ('KEY_F19', TPAS_EXIT, SMG$K_KEY_F19, SMG$SMASK_ADR),
: 288 P 0363 1 ('KEY_F20', TPAS_EXIT, SMG$K_KEY_F20, SMG$SMASK_ADR),
: 289 P 0364 1 ('KEY_LEFT_ARROW', TPAS_EXIT, SMG$K_KEY_LEFT_ARROW, SMG$SMASK_ADR),
: 290 P 0365 1 ('KEY_MINUS', TPAS_EXIT, SMG$K_KEY_MINUS, SMG$SMASK_ADR),
: 291 P 0366 1 ('KEY_PERIOD', TPAS_EXIT, SMG$K_KEY_PERIOD, SMG$SMASK_ADR),
: 292 P 0367 1 ('KEY_PF1', TPAS_EXIT, SMG$K_KEY_PF1, SMG$SMASK_ADR),
: 293 P 0368 1 ('KEY_PF2', TPAS_EXIT, SMG$K_KEY_PF2, SMG$SMASK_ADR),
: 294 P 0369 1 ('KEY_PF3', TPAS_EXIT, SMG$K_KEY_PF3, SMG$SMASK_ADR),
: 295 P 0370 1 ('KEY_PF4', TPAS_EXIT, SMG$K_KEY_PF4, SMG$SMASK_ADR),
: 296 P 0371 1 ('KEY_RIGHT_ARROW', TPAS_EXIT, SMG$K_KEY_RIGHT_ARROW, SMG$SMASK_ADR),
: 297 P 0372 1 ('KEY_UP_ARROW', TPAS_EXIT, SMG$K_KEY_UP_ARROW, SMG$SMASK_ADR),
: 298 P 0373 1 ('LABEL_F1', TPAS_EXIT, SMG$K_LABEL_F1, SMG$SMASK_ADR),
: 299 P 0374 1 ('LABEL_F2', TPAS_EXIT, SMG$K_LABEL_F2, SMG$SMASK_ADR),
: 300 P 0375 1 ('LABEL_F3', TPAS_EXIT, SMG$K_LABEL_F3, SMG$SMASK_ADR),
: 301 P 0376 1 ('LABEL_F4', TPAS_EXIT, SMG$K_LABEL_F4, SMG$SMASK_ADR),
: 302 P 0377 1 ('LABEL_F5', TPAS_EXIT, SMG$K_LABEL_F5, SMG$SMASK_ADR),
: 303 P 0378 1 ('LABEL_F6', TPAS_EXIT, SMG$K_LABEL_F6, SMG$SMASK_ADR),
: 304 P 0379 1 ('LABEL_F7', TPAS_EXIT, SMG$K_LABEL_F7, SMG$SMASK_ADR),
: 305 P 0380 1 ('LABEL_F8', TPAS_EXIT, SMG$K_LABEL_F8, SMG$SMASK_ADR),
: 306 P 0381 1 ('LABEL_F9', TPAS_EXIT, SMG$K_LABEL_F9, SMG$SMASK_ADR),
: 307 P 0382 1 ('LABEL_F10', TPAS_EXIT, SMG$K_LABEL_F10, SMG$SMASK_ADR),
: 308 P 0383 1 ('LABEL_F11', TPAS_EXIT, SMG$K_LABEL_F11, SMG$SMASK_ADR),
: 309 P 0384 1 ('LABEL_F12', TPAS_EXIT, SMG$K_LABEL_F12, SMG$SMASK_ADR),
: 310 P 0385 1 ('LABEL_F13', TPAS_EXIT, SMG$K_LABEL_F13, SMG$SMASK_ADR),
: 311 P 0386 1 ('LABEL_F14', TPAS_EXIT, SMG$K_LABEL_F14, SMG$SMASK_ADR),
: 312 P 0387 1 ('LABEL_F15', TPAS_EXIT, SMG$K_LABEL_F15, SMG$SMASK_ADR),
: 313 P 0388 1 ('LABEL_F16', TPAS_EXIT, SMG$K_LABEL_F16, SMG$SMASK_ADR),
: 314 P 0389 1 ('LABEL_F17', TPAS_EXIT, SMG$K_LABEL_F17, SMG$SMASK_ADR),

```

```

315 P 0390 1 ('LABEL_F18', TPAS_EXIT, ., SMG$K_LABEL_F18, SMG$SMASK_ADR),
316 P 0391 1 ('LABEL_F19', TPAS_EXIT, ., SMG$K_LABEL_F19, SMG$SMASK_ADR),
317 P 0392 1 ('LABEL_F20', TPAS_EXIT, ., SMG$K_LABEL_F20, SMG$SMASK_ADR),
318 P 0393 1 ('LEFT_T_CHAR', TPAS_EXIT, ., SMG$K_LEFT_T_CHAR, SMG$SMASK_ADR),
319 P 0394 1 ('LF_GRAPHIC', TPAS_EXIT, ., SMG$K_LF_GRAPHIC, SMG$SMASK_ADR),
320 P 0395 1 ('LIGHT_SCREEN', TPAS_EXIT, ., SMG$K_LIGHT_SCREEN, SMG$SMASK_ADR),
321 P 0396 1 ('LOWER_LEFT_CORNER', TPAS_EXIT, ., SMG$K_LOWER_LEFT_CORNER, SMG$SMASK_ADR),
322 P 0397 1 ('LOWER_RIGHT_CORNER', TPAS_EXIT, ., SMG$K_LOWER_RIGHT_CORNER, SMG$SMASK_ADR),
323 P 0398 1 ('NAME', TPAS_EXIT, SMG$K_NAME, SMG$SMASK_ADR),
324 P 0399 1 ('NEWLINE_CHAR', TPAS_EXIT, SMG$K_NEOLINE_CHAR, SMG$SMASK_ADR),
325 P 0400 1 ('PAD_CHAR', TPAS_EXIT, SMG$K_PAD_CHAR, SMG$SMASK_ADR),
326 P 0401 1 ('REQUEST_CURSOR_POSITION', TPAS_EXIT, SMG$K_REQUEST_CURSOR_POSITION, SMG$SMASK_ADR),
327 P 0402 1 ('RESTORE_CURSOR', TPAS_EXIT, ., SMG$K_RESTORE_CURSOR, SMG$SMASK_ADR),
328 P 0403 1 ('REVERSE_INDEX', TPAS_EXIT, ., SMG$K_REVERSE_INDEX, SMG$SMASK_ADR),
329 P 0404 1 ('RIGHT_T_CHAR', TPAS_EXIT, ., SMG$K_RIGHT_T_CHAR, SMG$SMASK_ADR),
330 P 0405 1 ('SAVE_CURSOR', TPAS_EXIT, SMG$K_SAVE_CURSOR, SMG$SMASK_ADR),
331 P 0406 1 ('SCROLL_FORWARD', TPAS_EXIT, ., SMG$K_SCROLL_FORWARD, SMG$SMASK_ADR),
332 P 0407 1 ('SCROLL_REVERSE', TPAS_EXIT, ., SMG$K_SCROLL_REVERSE, SMG$SMASK_ADR),
333 P 0408 1 ('SEL_ERASE_TO_END_DISPLAY', TPAS_EXIT, ., SMG$K_SEL_ERASE_TO_END_DISPLAY, SMG$SMASK_ADR),
334 P 0409 1 ('SEL_ERASE_TO_END_LINE', TPAS_EXIT, ., SMG$K_SEL_ERASE_TO_END_LINE, SMG$SMASK_ADR),
335 P 0410 1 ('SEL_ERASE_WHOLE_DISPLAY', TPAS_EXIT, ., SMG$K_SEL_ERASE_WHOLE_DISPLAY, SMG$SMASK_ADR),
336 P 0411 1 ('SEL_ERASE_WHOLE_LINE', TPAS_EXIT, ., SMG$K_SEL_ERASE_WHOLE_LINE, SMG$SMASK_ADR),
337 P 0412 1 ('SET_APPLICATION_KEYPAD', TPAS_EXIT, ., SMG$K_SET_APPLICATION_KEYPAD, SMG$SMASK_ADR),
338 P 0413 1 ('SET_CHAR_NOT_SEL_ERASE', TPAS_EXIT, ., SMG$K_SET_CHAR_NOT_SEL_ERASE, SMG$SMASK_ADR),
339 P 0414 1 ('SET_CHAR_SEL_ERASE', TPAS_EXIT, ., SMG$K_SET_CHAR_SEL_ERASE, SMG$SMASK_ADR),
340 P 0415 1 ('SET_CURSOR_ABS', TPAS_EXIT, ., SMG$K_SET_CURSOR_ABS, SMG$SMASK_ADR),
341 P 0416 1 ('SET_NUMERIC_KEYPAD', TPAS_EXIT, ., SMG$K_SET_NUMERIC_KEYPAD, SMG$SMASK_ADR),
342 P 0417 1 ('SET_SCROLL_REGION', TPAS_EXIT, ., SMG$K_SET_SCROLL_REGION, SMG$SMASK_ADR),
343 P 0418 1 ('SET_TAB', TPAS_EXIT, ., SMG$K_SET_TAB, SMG$SMASK_ADR),
344 P 0419 1 ('SINGLE_HIGH', TPAS_EXIT, ., SMG$K_SINGLE_HIGH, SMG$SMASK_ADR),
345 P 0420 1 ('TAB_CHAR', TPAS_EXIT, ., SMG$K_TAB_CHAR, SMG$SMASK_ADR),
346 P 0421 1 ('TOP_T_CHAR', TPAS_EXIT, ., SMG$K_TOP_T_CHAR, SMG$SMASK_ADR),
347 P 0422 1 ('TRUNCATION_ICON', TPAS_EXIT, ., SMG$K_TRUNCATION_ICON, SMG$SMASK_ADR),
348 P 0423 1 ('UNDERLINE_CHAR', TPAS_EXIT, ., SMG$K_UNDERLINE_CHAR, SMG$SMASK_ADR),
349 P 0424 1 ('UPPER_LEFT_CORNER', TPAS_EXIT, ., SMG$K_UPPER_LEFT_CORNER, SMG$SMASK_ADR),
350 P 0425 1 ('UPPER_RIGHT_CORNER', TPAS_EXIT, ., SMG$K_UPPER_RIGHT_CORNER, SMG$SMASK_ADR),
351 P 0426 1 ('VERTICAL_BAR', TPAS_EXIT, ., SMG$K_VERTICAL_BAR, SMG$SMASK_ADR),
352 P 0427 1 ('VT_GRAPHIC', TPAS_EXIT, ., SMG$K_VT_GRAPHIC, SMG$SMASK_ADR),
353 P 0428 1 ('WIDTH_NARROW', TPAS_EXIT, ., SMG$K_WIDTH_NARROW, SMG$SMASK_ADR),
354 P 0429 1 ('WIDTH_WIDE', TPAS_EXIT, ., SMG$K_WIDTH_WIDE, SMG$SMASK_ADR),
355 P 0430 1 ('PRIVATE_STR_1', TPAS_EXIT, ., SMG$K_PRIVATE_STR_1, SMG$SMASK_ADR),
356 P 0431 1 ('PRIVATE_STR_2', TPAS_EXIT, ., SMG$K_PRIVATE_STR_2, SMG$SMASK_ADR),
357 P 0432 1 ('PRIVATE_STR_3', TPAS_EXIT, ., SMG$K_PRIVATE_STR_3, SMG$SMASK_ADR),
358 P 0433 1 ('PRIVATE_STR_4', TPAS_EXIT, ., SMG$K_PRIVATE_STR_4, SMG$SMASK_ADR),
359 P 0434 1 ('PRIVATE_STR_5', TPAS_EXIT, ., SMG$K_PRIVATE_STR_5, SMG$SMASK_ADR),
360 P 0435 1 ('PRIVATE_STR_6', TPAS_EXIT, ., SMG$K_PRIVATE_STR_6, SMG$SMASK_ADR),
361 P 0436 1 ('PRIVATE_STR_7', TPAS_EXIT, ., SMG$K_PRIVATE_STR_7, SMG$SMASK_ADR),
362 P 0437 1 ('PRIVATE_STR_8', TPAS_EXIT, ., SMG$K_PRIVATE_STR_8, SMG$SMASK_ADR),
363 P 0438 1 ('PRIVATE_STR_9', TPAS_EXIT, ., SMG$K_PRIVATE_STR_9, SMG$SMASK_ADR),
364 P 0439 1 ('PRIVATE_STR_10', TPAS_EXIT, ., SMG$K_PRIVATE_STR_10, SMG$SMASK_ADR),
365 P 0440 1 ('TPAS_LAMBDA, TPAS_FAIL')
366 0441 1 );
367 0442 1 ;
368 0443 1 +;
369 0444 1 |+ Found a name, now look for the value. Skip over the intervening
370 0445 1 | equals sign. (If we get to '=', then we also have a capability
371 0446 1 | name/number to remember.)

```

: 372 0447 1 !-
: 373 0448 1
: 374 P 0449 1 \$STATE (EQUALS_STRING,
: 375 P 0450 1 ((END_OF_LINE), EQUALS_STRING, SMG\$NEXT_RECORD),
: 376 P 0451 1 ('=' STRING_CAP_VALUE, SMG\$STORE_CAP_MASK),
: 377 P 0452 1 (TPAS_SYMBOL, SMG\$SYNTAX_ERROR),
: 378 P 0453 1 (TPAS_ANY, , SMG\$SYNTAX_ERROR)
: 379 0454 1);
: 380 0455 1
: 381 0456 1 !+
: 382 0457 1 | We have found a capability name, now we need to decipher its value.
: 383 0458 1 | We found a string capability name so this better be a string value.
: 384 0459 1 !-
: 385 0460 1
: 386 P 0461 1 \$STATE (STRING_CAP_VALUE,
: 387 P 0462 1 ((END_OF_LINE), STRING_CAP_VALUE, SMG\$NEXT_RECORD),
: 388 P 0463 1 ((STRING_CAP), BEGIN_SCAN),
: 389 P 0464 1 (TPAS_SYMBOL, , SMG\$SYNTAX_ERROR),
: 390 P 0465 1 (TPAS_ANY, , SMG\$SYNTAX_ERROR)
: 391 0466 1);
: 392 0467 1
: 393 P 0468 1 \$STATE (COMMA,
: 394 P 0469 1 (',', TPAS_EXIT, SMG\$BLANKS_OFF),
: 395 P 0470 1 (TPAS_BLANK, COMMA, SMG\$BLANKS_OFF),
: 396 P 0471 1 ((END_OF_LINE), TPAS_EXIT, SMG\$NEXT_RECORD),
: 397 P 0472 1 (TPAS_ANY, , SMG\$SYNTAX_ERROR)
: 398 0473 1);
: 399 0474 1
: 400 0475 1 !+
: 401 0476 1 | A string capability requires further parsing. There may be an embedded
: 402 0477 1 | special character, an FAO call, or arithmetic. A matching quote indicates
: 403 0478 1 | the end of the string value, which should be stored in TERMTABLE.EXE.
: 404 0479 1 !-
: 405 0480 1
: 406 P 0481 1 \$STATE (STRING_CAP,
: 407 P 0482 1 (SINGLE_QUOTE, SINGLE_QUOTE_STRING, SMG\$BLANKS_ON),
: 408 P 0483 1 (DOUBLE_QUOTE, DOUBLE_QUOTE_STRING, SMG\$BLANKS_ON),
: 409 P 0484 1 (TPAS_ANY, TPAS_FAIL)
: 410 0485 1);
: 411 0486 1
: 412 P 0487 1 \$STATE (SINGLE_QUOTE_STRING,
: 413 P 0488 1 ((UNDERSCORE), SINGLE_QUOTE_STRING),
: 414 P 0489 1 ('\$', SINGLE_QUOTE_STRING, CONVERT_ESCAPE),
: 415 P 0490 1 ((CARROT), SINGLE_QUOTE_STRING),
: 416 P 0491 1 ((FAO_STRING), SINGLE_QUOTE_STRING),
: 417 P 0492 1 ((ARITH_CAP), SINGLE_QUOTE_STRING),
: 418 P 0493 1 (SINGLE_QUOTE, COMMA, SMG\$COPY_CAP),
: 419 P 0494 1 (TPAS_BLANK, SINGLE_QUOTE_STRING, SMG\$SAVE_TOKEN_STRING),
: 420 P 0495 1 (DOUBLE_QUOTE, SMG\$SYNTAX_ERROR),
: 421 P 0496 1 (TPAS_ANY, SINGLE_QUOTE_STRING, SMG\$SAVE_TOKEN_STRING),
: 422 P 0497 1 (TPAS_LAMBDA, TPAS_EXIT)
: 423 0498 1);
: 424 0499 1
: 425 P 0500 1 \$STATE (DOUBLE_QUOTE_STRING,
: 426 P 0501 1 ((UNDERSCORE), DOUBLE_QUOTE_STRING),
: 427 P 0502 1 ('\$', DOUBLE_QUOTE_STRING, CONVERT_ESCAPE),
: 428 P 0503 1 ((CARROT), DOUBLE_QUOTE_STRING),

```
: 429      P 0504 1 ((FAO STRING), DOUBLE_QUOTE STRING),  
430      P 0505 1 ((ARITH_CAP), DOUBLE_QUOTE STRING),  
431      P 0506 1 (DOUBLE_QUOTE, COMMA, SMG$COPY_CAP),  
432      P 0507 1 (TPAS_BLANK, DOUBLE_QUOTE STRING, SMG$SAVE_TOKEN_STRING),  
433      P 0508 1 (SINGLE_QUOTE, SMG$SYNTAX_ERROR),  
434      P 0509 1 (TPAS_ANY, DOUBLE_QUOTE STRING, SMG$SAVE_TOKEN_STRING),  
435      P 0510 1 (TPAS_LAMBDA, TPAS_EXIT)  
436      0511 1 );  
437      0512 1  
438      0513 1 +  
439      0514 1 | Underscore is used to insert special characters into a string as  
440      0515 1 | normal text. If this underscore is followed by a special character,  
441      0516 1 | skip over it and just store the special character.  
442      0517 1 -  
443      0518 1  
444      P 0519 1 $STATE(UNDERSCORE,  
445      P 0520 1 (' ', UNDERSCORE2),  
446      P 0521 1 (TPAS_LAMBDA, TPAS_FAIL)  
447      0522 1 );  
448      0523 1  
449      P 0524 1 $STATE(UNDERSCORE2,  
450      P 0525 1 ('$', TPAS_EXIT, INSERT_DOLLAR),  
451      P 0526 1 ('^', TPAS_EXIT, INSERT_CARROT),  
452      P 0527 1 ('!', TPAS_EXIT, INSERT_EXCLAMATION),  
453      P 0528 1 ('(', TPAS_EXIT, INSERT_PARENTHESIS),  
454      P 0529 1 (TPAS_LAMBDA, TPAS_FAIL)  
455      0530 1 );  
456      0531 1  
457      0532 1 +  
458      0533 1 | If we find a ^, advance to the next character and call the conversion  
459      0534 1 | routine. The conversion routine will complain if this is a control  
460      0535 1 | sequence we don't know about.  
461      0536 1 -  
462      0537 1  
463      P 0538 1 $STATE(CARROT,  
464      P 0539 1 ('^', CONTROL_CHAR),  
465      P 0540 1 (TPAS_LAMBDA, TPAS_FAIL)  
466      0541 1 );  
467      0542 1  
468      P 0543 1 $STATE(CONTROL_CHAR,  
469      P 0544 1 (TPAS_ANY, TPAS_EXIT, CONVERT_CONTROL)  
470      0545 1 );  
471      0546 1  
472      0547 1 +  
473      0548 1 | Check for a FAO directive. Make sure it's one we understand.  
474      0549 1 -  
475      P 0550 1 $STATE(FAO_STRING,  
476      P 0551 1 ('!', SMG$SAVE_TOKEN_STRING),  
477      P 0552 1 (TPAS_LAMBDA, TPAS_FAIL)  
478      0553 1 );  
479      0554 1  
480      P 0555 1 $STATE(,  
481      P 0556 1 ('U', SMG$SAVE_TOKEN_STRING),  
482      P 0557 1 (TPAS_SYMBOL, TPAS_FAIL, INVALID_DIRECTIVE),  
483      P 0558 1 (TPAS_ANY, TPAS_FAIL, INVALID_DIRECTIVE)  
484      0559 1 );  
485      0560 1
```

```
486 P 0561 1 $STATE {,  
487 P 0562 1 ('L', TPAS_EXIT, SMG$SAVE_TOKEN_STRING, SMG$K_FA0_STRING, SMG$MASK_ADR),  
488 P 0563 1 (TPAS_SYMBOL, TPAS_FAIL, INVALID_DIRECTIVE),  
489 P 0564 1 (TPAS_ANY, TPAS_FAIL, INVALID_DIRECTIVE)  
490 0565 1 );  
491 0566 1 ;  
492 0567 1 +  
493 0568 1 Parse arithmetic string. We can't complete arithmetic processing until  
494 0569 1 run-time, but here we pre-compile as much as possible. Arithmetic  
495 0570 1 expressions are stored as number, op, number, op, number ... etc.  
496 0571 1 -  
497 0572 1 ;  
498 P 0573 1 $STATE (ARITH_CAP,  
499 P 0574 1 (LEFT_PAREN, ARITH_CAP2, INIT_ARITH_CAP),  
500 P 0575 1 (TPAS_BLANK, ARITH_CAP),  
501 P 0576 1 (TPAS_LAMBDA, TPAS_FAIL)  
502 0577 1 );  
503 0578 1 ;  
504 P 0579 1 $STATE (ARITH_CAP2,  
505 P 0580 1 ((OPERAND, TPAS_EXIT),  
506 P 0581 1 (TPAS_LAMBDA, TPAS_FAIL)) ; errors handled in OPERAND  
507 0582 1 ; return w/out consuming next token  
508 0583 1 ;  
509 P 0584 1 $STATE (OPERATOR,  
510 P 0585 1 ('+', OPERAND, SMG$K_ADD, SMG$MASK_ADR),  
511 P 0586 1 ('-', OPERAND, SMG$K_SUBTRACT, SMG$MASK_ADR),  
512 P 0587 1 ('*', OPERAND, SMG$K_MULTIPLY, SMG$MASK_ADR),  
513 P 0588 1 ('/', OPERAND, SMG$K_DIVIDE, SMG$MASK_ADR),  
514 P 0589 1 (RIGHT_PAREN, TPAS_EXIT, END_ARITH_CAP),  
515 P 0590 1 (TPAS_ANY, , EXPRESSION_ERROR)  
516 0591 1 );  
517 0592 1 ;  
518 P 0593 1 $STATE (OPERAND,  
519 P 0594 1 (TPAS_DECIMAL, OPERATOR, STORE_OPERAND),  
520 P 0595 1 ((SUBSTITUTION), OPERATOR),  
521 P 0596 1 (TPAS_SYMBOL, , EXPRESSION_ERROR),  
522 P 0597 1 (TPAS_ANY, , EXPRESSION_ERROR)  
523 0598 1 );  
524 0599 1 ;  
525 P 0600 1 $STATE (SUBSTITUTION,  
526 P 0601 1 ('%'),  
527 P 0602 1 (TPAS_LAMBDA, TPAS_FAIL)  
528 0603 1 );  
529 0604 1 ;  
530 0605 1 ! % should be followed by a number  
531 P 0606 1 $STATE {  
532 P 0607 1 (TPAS_DECIMAL, TPAS_EXIT, STORE_SUBSTITUTION),  
533 P 0608 1 (TPAS_ANY, , EXPRESSION_ERROR)  
534 0609 1 );  
535 0610 1 ;  
536 0611 1 ;
```

```

538      0612 1 %SBTTL 'CONVERT_CONTROL - Convert ^ to control character'
539      0613 1 ROUTINE CONVERT_CONTROL =
540      0614 1
541      0615 1 ++
542      0616 1 FUNCTIONAL DESCRIPTION:
543      0617 1
544      0618 1 Stores a control character in the current capability string in
545      0619 1 TERMTABLE.EXE.
546      0620 1
547      0621 1 CALLING SEQUENCE:
548      0622 1
549      0623 1     status = CONVERT_CONTROL ()
550      0624 1
551      0625 1 FORMAL PARAMETERS:
552      0626 1
553      0627 1     NONE
554      0628 1
555      0629 1 IMPLICIT INPUTS:
556      0630 1
557      0631 1     AP      Points to TPARSE parameter block
558      0632 1
559      0633 1 IMPLICIT OUTPUTS:
560      0634 1
561      0635 1     NONE
562      0636 1
563      0637 1 COMPLETION STATUS:
564      0638 1
565      0639 1     SSS_NORMAL
566      0640 1
567      0641 1 SIDE EFFECTS:
568      0642 1
569      0643 1
570      0644 1
571      0645 2
572      0646 2 BEGIN
573      0647 2     BUILTIN
574      0648 2     CALLG,
575      0649 2     AP;
576      0650 2     MAP
577      0651 2     AP : REF BLOCK [,BYTE];
578      0652 2     BIND
579      0653 2     LOCAL
580      0654 2     CAP_PTRS = .AP [PARAM_L_CUR_TERM_DEF] : VECTOR [,WORD];
581      0655 2     CONTROL_CHAR : BYTE,           ! char following ctrl
582      0656 2     MOVE_CHAR : BYTE;           ! code to place in cap string
583      0657 2
584      0658 2     The NAME capability should have preceeded this. Complain if it
585      0659 2     didn't.
586      0660 2
587      0661 2
588      0662 2     IF CAP_PTRS EQL 0
589      0663 2     THEN
590      0664 2     SIGNAL_STOP (SMGS_MISTERNAM);
591      0665 2
592      0666 2
593      0667 2     Control must be followed by a valid character in order to constitute
594      0668 2     a control code. The character can be shifted or not (lower or upper

```

```

: 595 0669 2 | case).
: 596 0670 2 | Since we have already seen a control character, we can translate on
: 597 0671 2 | the second character in the control string.
: 598 0672 2 |
: 599 0673 2 |
: 600 0674 2 | CONTROL_CHAR = .(AP [TPASL_TOKENPTR]) <0,8>;
: 601 0675 2 |
: 602 0676 2 | A control character is in the range 00 - 1F (hex) but is expressed as
: 603 0677 2 | 40 - 5F (hex). Strip off extra bit to get the control character.
: 604 0678 2 |
: 605 0679 2 |
: 606 0680 2 | MOVE_CHAR = .CONTROL_CHAR AND NOT (%X'40');
: 607 0681 2 |
: 608 0682 2 |
: 609 0683 2 | We now have the byte which must be moved into the capability string.
: 610 0684 2 | This may not be the first byte of the string - check for saved characters
: 611 0685 2 | which have not been copied yet, and append in the control code.
: 612 0686 2 |
: 613 0687 2 |
: 614 0688 2 | CALLG (.AP, SMG$FLUSH_SAVED_BUFFER); ! copy saved string
: 615 0689 2 |
: 616 0690 2 | AP [PARAM_L_SAVED_TOKENCNT] = 1;
: 617 0691 2 | AP [PARAM_L_SAVED_TOKENSTR] = MOVE_CHAR; ! control char is now 'saved'
: 618 0692 2 |
: 619 0693 2 | CALLG (.AP, SMG$FLUSH_SAVED_BUFFER); ! append control char
: 620 0694 2 |
: 621 0695 2 | RETURN (SS$_NORMAL);
: 622 0696 2 |
: 623 0697 1 | END: ! end of routine CONVERT_CONTROL

```

.TITLE SMG\$STRING_TABLES TPARSE tables for string capabilities

.IDENT \1-003\

.PSECT _LIB\$KEY1\$,NOWRT, SHR, PIC,1

	00000	:TPASKEYSTO		
	U.29:	.BLKB	0	
44 4E 45	00000	:TPASKEYST		
	U.31:	.ASCII	\END\	
FF	00003	.BYTE	-1	
	00004	:TPASKEYSTO		
	U.34:	.BLKB	0	
4E 41 45 4C 4F 4F 42	00004	:TPASKEYST		
	U.36:	.ASCII	\BOOLEAN\	
FF	00008	.BYTE	-1	
	0000C	:TPASKEYSTO		
	U.39:	.BLKB	0	
43 49 52 45 4D 55 4E	0000C	:TPASKEYST		
	U.41:	.ASCII	\NUMERIC\	
FF	00013	.BYTE	-1	
	00014	:TPASKEYSTO		
	U.44:	.BLKB	0	
47 4E 49 52 54 53	00014	:TPASKEYST		
	U.46:	.ASCII	\STRING\	
FF	0001A	.BYTE	-1	

45 52 49 55 51 45 52	0001B	;TPASKEYSTO U.49: .BLKB 0	;
FF	0001B	;TPASKEYST U.51: .ASCII \REQUIRE\	;
45 4D 41 4E	00022	.BYTE -1	55
FF	00023	;TPASKEYSTO U.55: .BLKB 0	;
FF	00027	;TPASKEYST U.57: .ASCII \NAME\	;
FF	00028	.BYTE -1	49
45 54 41 4E 52 45 54	00029	;TPASKEYFILL U.63: .BYTE -1	;
4C 41 5F 4E 49 47 45	00029	;TPASKEYSTO U.64: .BLKB 0	;
42	00029	;TPASKEYST U.66: .ASCII \BEGIN_ALTERNATE_CHAR\	;
52 41 48 43	00038	.BYTE -1	;
FF	0003D	;TPASKEYSTO U.71: .BLKB 0	;
4B 4E 49 4C 42 5F	0003E	;TPASKEYST U.73: .ASCII \BEGIN_BLINK\	;
4E 49 47 45 42	00049	.BYTE -1	;
FF	0004A	;TPASKEYSTO U.78: .BLKB 0	;
44 4C 4F 42 5F	0004A	;TPASKEYST U.80: .ASCII \BEGIN_BOLD\	;
4E 49 47 45 42	00054	.BYTE -1	;
FF	00055	;TPASKEYSTO U.85: .BLKB 0	;
4F 4D 5F 45 54	00055	;TPASKEYST U.87: .ASCII \BEGIN_DELETE_MODE\	;
45 44	00064	.BYTE -1	;
FF	00066	;TPASKEYSTO U.92: .BLKB 0	;
4F 4D 5F 54 52	00067	;TPASKEYST U.94: .ASCII \BEGIN_INSERT_MODE\	4E
45 53 4E 49 5F	00076	.BYTE -1	;
4E 49 47 45 42	00078	;TPASKEYSTO U.99: .BLKB 0	;
52 41 48 43	00079	;TPASKEYST U.101: .ASCII \BEGIN_LINE_DRAWING_CHAR\	;
5F 47 4E 49	00088	.BYTE -1	;
FF	00090	;TPASKEYSTO U.106: .BLKB 0	41
45 52 5F 4C 41	00091	;TPASKEYST U.108: .ASCII \BEGIN_NORMAL_RENDERING\	;
4D 52 4F 4E 5F	000A0	.BYTE -1	;
4E 4F 49 54 49	000A7	;TPASKEYSTO U.113: .BLKB 0	51
44 4E	000A8	;TPASKEYST U.115: .ASCII \BEGIN_REVERSE\	;
45 53 52 45 56	000AB	.BYTE -1	;
45 52 5F 4E 49	FF	;TPASKEYST U.115: .ASCII \BEGIN_REVERSE\	;
47 45 42	00085	.BYTE -1	;

52 4F 43 53 52 45 44 4E 55 5F 4E 49 47 45 42	00086 ;TPASKEYST U.120: BLKB 0	;
45 000C5 FF 000C6 000C7	;TPASKEYST U.122: .ASCII \BEGIN_UNDERSCORE\ .BYTE -1	;
52 41 48 43 5F 54 5F 4D 4F 54 54 4F 42 000C7 FF 000D4 000D5	;TPASKEYST U.127: BLKB 0 U.129: .ASCII \BOTTOM_T_CHAR\ .BYTE -1	;
42 41 54 5F 52 41 45 4C 43 000D5 FF 000DE 000DF	;TPASKEYST U.134: BLKB 0 U.136: .ASCII \CLEAR_TAB\ .BYTE -1	;
43 49 48 50 41 52 47 5F 52 43 000DF FF 000E9 000EA	;TPASKEYST U.141: BLKB 0 U.143: .ASCII \CR_GRAPHIC\ .BYTE -1	;
52 41 48 43 5F 53 53 4F 52 43 000EA FF 000F4 000F5	;TPASKEYST U.148: BLKB 0 U.150: .ASCII \CROSS_CHAR\ .BYTE -1	;
4E 57 4F 44 5F 52 4F 53 52 55 43 000F5 FF 00100 00101	;TPASKEYST U.155: BLKB 0 U.157: .ASCII \CURSOR_DOWN\ .BYTE -1	;
54 46 45 4C 5F 52 4F 53 52 55 43 00101 FF 0010C 0010D	;TPASKEYST U.162: BLKB 0 U.164: .ASCII \CURSOR_LEFT\ .BYTE -1	;
4E 49 4C 5F 54 58 45 4E 5F 52 4F 53 52 55 43 0010D 45 0011C FF 0011D 0011E	;TPASKEYST U.169: BLKB 0 U.171: .ASCII \CURSOR_NEXT_LINE\ .BYTE -1	;
4E 4F 49 54 49 53 4F 50 5F 52 4F 53 52 55 43 0011E 54 52 4F 50 45 52 5F 0012D FF 00134 00135	;TPASKEYST U.176: BLKB 0 U.178: .ASCII \CURSOR_POSITION_REPORT\ .BYTE -1	;
49 44 45 45 43 45 52 50 5F 52 4F 53 52 55 43 00135 45 4E 49 4C 5F 47 4E 00144 FF 0014B 0014C	;TPASKEYST U.183: BLKB 0 U.185: .ASCII \CURSOR_PRECEDING_LINE\ .BYTE -1	;
54 48 47 49 52 5F 52 4F 53 52 55 43 0014C	;TPASKEYST U.190: BLKB 0	;

FF	00158	U.192: .ASCII \CURSOR_RIGHT\	;
	00159	;TPASKEY\$TO .BYTE -1	;
50	55	5F 52 4F 53 52 55 43 00159 ;TPASKEY\$T .BLKB 0	;
		U.197: .BLKB 0	;
FF	00162	U.199: .ASCII \CURSOR_UP\	;
	00163	;TPASKEY\$TO .BYTE -1	;
4E	45	45 52 43 53 5F 48 52 41 44 00163 ;TPASKEY\$T .BLKB 0	;
		U.204: .BLKB 0	;
FF	0016E	U.206: .ASCII \DARK_SCREEN\	;
	0016F	;TPASKEY\$TO .BYTE -1	;
52	41	48 43 5F 45 54 45 4C 45 44 0016F ;TPASKEY\$T .BLKB 0	;
		U.211: .BLKB 0	;
FF	0017A	U.213: .ASCII \DELETE_CHAR\	;
	0017B	;TPASKEY\$TO .BYTE -1	;
45	4E	49 4C 5F 45 54 45 4C 45 44 0017B ;TPASKEY\$T .BLKB 0	;
		U.218: .BLKB 0	;
FF	00186	U.220: .ASCII \DELETE_LINE\	;
	00187	;TPASKEY\$TO .BYTE -1	;
54	55	42 48 52 54 54 41 5F 45 43 49 56 45 44 00187 ;TPASKEY\$T .BLKB 0	;
		U.225: .BLKB 0	;
53	45	00187 ;TPASKEY\$T .ASCII \DEVICE_ATTRIBUTES\	;
FF	00196	;	;
	00198	TPASKEY\$TO .BYTE -1	;
54	4F	00199 ;TPASKEY\$T .BLKB 0	;
		U.232: .BLKB 0	;
4D	4F	54 001A8 ;TPASKEY\$T .ASCII \DOUBLE_HIGH_BOTTOM\	;
		U.234: .ASCII \DOUBLE_HIGH_BOTTOM\	;
FF	001AB	;	;
	001AC	TPASKEY\$TO .BYTE -1	;
50	4F	54 5F 48 47 49 48 5F 45 4C 42 55 4F 44 001AC ;TPASKEY\$T .BLKB 0	;
		U.239: .BLKB 0	;
FF	001BB	U.241: .ASCII \DOUBLE_HIGH_TOP\	;
	001BC	;TPASKEY\$TO .BYTE -1	;
45	44	49 57 5F 45 4C 42 55 4F 44 001BC ;TPASKEY\$T .BLKB 0	;
		U.246: .BLKB 0	;
FF	001C7	U.248: .ASCII \DOUBLE_WIDE\	;
	001C8	;TPASKEY\$TO .BYTE -1	;
45	54	41 43 49 4C 50 55 44 001C8 ;TPASKEY\$T .BLKB 0	;
		U.253: .BLKB 0	;
FF	001D1	U.255: .ASCII \DUPLICATE\	;
	001D2	;TPASKEY\$TO .BYTE -1	;
43	5F	45 54 41 4E 52 45 54 4C 41 5F 44 4E 45 001D2 ;TPASKEY\$T .BLKB 0	;
		U.260: .BLKB 0	;
52	41	48 001E1 ;TPASKEY\$T .ASCII \END_ALTERNATE_CHAR\	;
		U.262: .ASCII \END_ALTERNATE_CHAR\	;
FF	001E4	;	;
	001E5	TPASKEY\$TO .BYTE -1	;
		U.267: .BLKB 0	;

SMG\$STRING_TABL TPARSE tables for string capabilities
1-003 CONVERT_CONTROL - Convert * to control

N 9

16-Sep-1984 01:22:35
14-Sep-1984 13:10:04

16-Sep-1984 01:22:39

VAX-11 Bliss-32 V4.0-742
[SMGRTL.SRC]SMGSTRTAB.B32;1

Page 16
(4)

48	4E	49	4C	42	5F	44	4E	45	001E5	TPASKEYST
							FF	001EE	U.269: .ASCII	\END_BLINK\
							001EF	001EF	:TPASKEYST0	-1
								001EF	U.274: .BLKB	0
								001F7	:TPASKEYST	
							FF	001F8	U.276: .ASCII	\END_BOLD\
							001F8	001F8	:TPASKEYST0	-1
							FF	001F8	U.281: .BLKB	0
							00207	00207	:TPASKEYST	
							00208	00208	U.283: .ASCII	\END_DELETE_MODE\
							FF	00217	:TPASKEYST0	-1
							00218	00218	U.288: .BLKB	0
							FF	00218	:TPASKEYST	
							00218	00218	U.290: .ASCII	\END_INSERT_MODE\
							FF	00218	:TPASKEYST0	-1
							FF	00227	U.295: .BLKB	0
							FF	00227	:TPASKEYST	
							FF	0022D	U.297: .ASCII	\END_LINE_DRAWING_CHAR\
							FF	0022D	:TPASKEYST0	-1
							FF	0022E	U.302: .BLKB	0
							FF	0022E	:TPASKEYST	
							FF	00239	U.304: .ASCII	\END_REVERSE\
							FF	00239	:TPASKEYST0	-1
							FF	0023A	U.309: .BLKB	0
							FF	0023A	:TPASKEYST	
							FF	00248	U.311: .ASCII	\END_UNDERSCORE\
							FF	00248	:TPASKEYST0	-1
							FF	00249	U.316: .BLKB	0
							FF	00249	:TPASKEYST	
							FF	00249	U.318: .ASCII	\ERASE_TO_END_DISPLAY\
							FF	00258	:TPASKEYST0	-1
							FF	0025D	U.323: .BLKB	0
							FF	0025E	:TPASKEYST	
							FF	0025E	U.325: .ASCII	\ERASE_TO_END_LINE\
							FF	0026D	:TPASKEYST0	-1
							FF	0026F	U.330: .BLKB	0
							FF	00270	:TPASKEYST	
							FF	00270	U.332: .ASCII	\ERASE_WHOLE_DISPLAY\
							FF	0027F	:TPASKEYST0	-1
							FF	00283	U.337: .BLKB	0
							FF	00284	:TPASKEYST	
							FF	00284	U.339: .ASCII	\ERASE_WHOLE_LINE\

	FF	00294	.BYTE	-1													
		00295	;TPASKEY\$TO														
43	49	48	50	41	52	47	5F	46	46	00295	U.344: .BLKB	0					
											U.346: .ASCII	\FF_GRAPHIC\					
											FF	0029F	.BYTE	-1			
											002A0	;TPASKEY\$TO					
											U.351: .BLKB	0					
45	4D	4F	48	.002A0	;TPASKEY\$T	U.353: .ASCII	\HOME\										
						FF	002A4	.BYTE	-1								
						002A5	;TPASKEY\$TO										
52	41	42	5F	4C	41	54	4E	4F	5A	49	52	4F	48	002A5	U.358: .BLKB	0	
											U.360: .ASCII	\HORIZONTAL_BAR\					
						FF	002B3	.BYTE	-1								
						002B4	;TPASKEY\$TO										
43	49	48	50	41	52	47	5F	54	48	002B4	U.365: .BLKB	0					
										U.367: .ASCII	\HT_GRAPHIC\						
						FF	002BE	.BYTE	-1								
						002BF	;TPASKEY\$TO										
						U.372: .BLKB	0										
58	45	44	4E	49	002BF	;TPASKEY\$T	U.374: .ASCII	\INDEX\									
						FF	002C4	.BYTE	-1								
						002C5	;TPASKEY\$TO										
47	4E	49	52	54	53	5F	54	49	4E	49	002C5	U.379: .BLKB	0				
										U.381: .ASCII	\INIT_STRING\						
						FF	002D0	.BYTE	-1								
						002D1	;TPASKEY\$TO										
52	41	48	43	5F	54	52	45	53	4E	49	002D1	U.386: .BLKB	0				
										U.388: .ASCII	\INSERT_CHAR\						
						FF	002DC	.BYTE	-1								
						002DD	;TPASKEY\$TO										
45	4E	49	4C	5F	54	52	45	53	4E	49	002DD	U.393: .BLKB	0				
										U.395: .ASCII	\INSERT_LINE\						
						FF	002E8	.BYTE	-1								
						002E9	;TPASKEY\$TO										
44	41	50	5F	54	52	45	53	4E	49	002E9	U.400: .BLKB	0					
										U.402: .ASCII	\INSERT_PAD\						
						FF	002F3	.BYTE	-1								
						002F4	;TPASKEY\$TO										
						U.407: .BLKB	0										
30	5F	59	45	4B	002F4	;TPASKEY\$T	U.409: .ASCII	\KEY_0\									
						FF	002F9	.BYTE	-1								
						002FA	;TPASKEY\$TO										
31	5F	59	45	4B	002FA	;TPASKEY\$T	U.414: .BLKB	0									
						FF	002FF	.BYTE	-1								
						00300	;TPASKEY\$TO										

32	5F	59	45	4B	00300	U.421: .BLKB	0	;									
				FF	00305	U.423: .ASCII	\KEY_2\	;									
					00306	:TPASKEY\$TO		;									
33	5F	59	45	4B	00306	U.428: .BLKB	0	;									
				FF	00308	U.430: .ASCII	\KEY_3\	;									
					0030C	:TPASKEY\$TO		;									
34	5F	59	45	4B	0030C	U.435: .BLKB	0	;									
				FF	00311	U.437: .ASCII	\KEY_4\	;									
					00312	:TPASKEY\$TO		;									
35	5F	59	45	4B	00312	U.442: .BLKB	0	;									
				FF	00317	U.444: .ASCII	\KEY_5\	;									
					00318	:TPASKEY\$TO		;									
36	5F	59	45	4B	00318	U.449: .BLKB	0	;									
				FF	0031D	U.451: .ASCII	\KEY_6\	;									
					0031E	:TPASKEY\$TO		;									
37	5F	59	45	4B	0031E	U.456: .BLKB	0	;									
				FF	00323	U.458: .ASCII	\KEY_7\	;									
					00324	:TPASKEY\$TO		;									
38	5F	59	45	4B	00324	U.463: .BLKB	0	;									
				FF	00329	U.465: .ASCII	\KEY_8\	;									
					0032A	:TPASKEY\$TO		;									
39	5F	59	45	4B	0032A	U.470: .BLKB	0	;									
				FF	0032F	U.472: .ASCII	\KEY_9\	;									
					00330	:TPASKEY\$TO		;									
45	43	41	50	53	4B	43	41	42	5F	59	45	4B	00330	U.477: .BLKB	0	;	
					FF	0033D	U.479: .ASCII	\KEY_BACKSPACE\	;								
						0033E	:TPASKEY\$TO		;								
41	4D	4D	4F	43	5F	59	45	4B	0033E	U.484: .BLKB	0	;					
				FF	00347	U.486: .ASCII	\KEY_COMMAS\	;									
					00348	:TPASKEY\$TO		;									
57	4F	52	52	41	5F	4E	57	4F	44	5F	59	45	4B	00348	U.491: .BLKB	0	;
				FF	00356	U.493: .ASCII	\KEY_DOWN_ARROW\	;									
					00357	:TPASKEY\$TO		;									
31	45	5F	59	45	4B	00357	U.498: .BLKB	0	;								

FF	0035D	U.500: .ASCII .BYTE	\KEY_E1\	:
	0035E	;TPA\$KEY\$TO		:
32	45	5F 59 45 4B 0035E	U.505: .BLKB	0
			;TPA\$KEY\$T	:
			U.507: .ASCII .BYTE	\KEY_E2\
			00364	:
			00365	;TPA\$KEY\$TO
33	45	5F 59 45 4B 00365	U.512: .BLKB	0
			;TPA\$KEY\$T	:
			U.514: .ASCII .BYTE	\KEY_E3\
			0036B	:
			0036C	;TPA\$KEY\$TO
34	45	5F 59 45 4B 0036C	U.519: .BLKB	0
			;TPA\$KEY\$T	:
			U.521: .ASCII .BYTE	\KEY_E4\
			00372	:
			00373	;TPA\$KEY\$TO
35	45	5F 59 45 4B 00373	U.526: .BLKB	0
			;TPA\$KEY\$T	:
			U.528: .ASCII .BYTE	\KEY_E5\
			00379	:
			0037A	;TPA\$KEY\$TO
36	45	5F 59 45 4B 0037A	U.533: .BLKB	0
			;TPA\$KEY\$T	:
			U.535: .ASCII .BYTE	\KEY_E6\
			00380	:
			00381	;TPA\$KEY\$TO
52	45	54 4E 45 5F 59 45 4B 00381	U.540: .BLKB	0
			;TPA\$KEY\$T	:
			U.542: .ASCII .BYTE	\KEY_ENTER\
			0038A	:
			0038B	;TPA\$KEY\$TO
31	46	5F 59 45 4B 0038B	U.547: .BLKB	0
			;TPA\$KEY\$T	:
			U.549: .ASCII .BYTE	\KEY_F1\
			00391	:
			00392	;TPA\$KEY\$TO
32	46	5F 59 45 4B 00392	U.554: .BLKB	0
			;TPA\$KEY\$T	:
			U.556: .ASCII .BYTE	\KEY_F2\
			00398	:
			00399	;TPA\$KEY\$TO
33	46	5F 59 45 4B 00399	U.561: .BLKB	0
			;TPA\$KEY\$T	:
			U.563: .ASCII .BYTE	\KEY_F3\
			0039F	:
			003A0	;TPA\$KEY\$TO
34	46	5F 59 45 4B 003A0	U.568: .BLKB	0
			;TPA\$KEY\$T	:
			U.570: .ASCII .BYTE	\KEY_F4\
			003A6	:
			003A7	;TPA\$KEY\$TO
35	46	5F 59 45 4B 003A7	U.575: .BLKB	0
			;TPA\$KEY\$T	:
			U.577: .ASCII .BYTE	\KEY_F5\
			003AD	:

36	46	5F	59	45	4B	003AE	:TPASKEYSTO		
						U.582:	:BLKB	0	
						U.584:	.ASCII	\KEY_F6\	
					FF	003B4	:BYTE	-1	
						003B5	:TPASKEYSTO		
						U.589:	:BLKB	0	
37	46	5F	59	45	4B	003B5	:TPASKEYST		
						U.591:	.ASCII	\KEY_F7\	
					FF	003BB	:BYTE	-1	
						003BC	:TPASKEYSTO		
						U.596:	:BLKB	0	
38	46	5F	59	45	4B	003BC	:TPASKEYST		
					FF	003C2	:BYTE	-1	
						003C3	:TPASKEYSTO		
						U.603:	:BLKB	0	
39	46	5F	59	45	4B	003C3	:TPASKEYST		
					FF	003C9	:BYTE	-1	
						003CA	:TPASKEYSTO		
						U.610:	:BLKB	0	
30	31	46	5F	59	45	4B	003CA	:TPASKEYST	
					FF	003D1	:BYTE	-1	
						003D2	:TPASKEYSTO		
						U.617:	:BLKB	0	
31	31	46	5F	59	45	4B	003D2	:TPASKEYST	
					FF	003D9	:BYTE	-1	
						003DA	:TPASKEYSTO		
						U.624:	:BLKB	0	
32	31	46	5F	59	45	4B	003DA	:TPASKEYST	
					FF	003E1	:BYTE	-1	
						003E2	:TPASKEYSTO		
						U.631:	:BLKB	0	
33	31	46	5F	59	45	4B	003E2	:TPASKEYST	
					FF	003E9	:BYTE	-1	
						003EA	:TPASKEYSTO		
						U.638:	:BLKB	0	
34	31	46	5F	59	45	4B	003EA	:TPASKEYST	
					FF	003F1	:BYTE	-1	
						003F2	:TPASKEYSTO		
						U.645:	:BLKB	0	
35	31	46	5F	59	45	4B	003F2	:TPASKEYST	
					FF	003F9	:BYTE	-1	
						003FA	:TPASKEYSTO		
						U.652:	:BLKB	0	
36	31	46	5F	59	45	4B	003FA	:TPASKEYST	
					FF	00401	:BYTE	-1	
						00402	:TPASKEYSTO		
						U.659:	:BLKB	0	

SMG\$STRING_TBL TPARSE tables for string capabilities
1-003 CONVERT_CONTROL - Convert ^ to control

F 10

16-Sep-1984 01:22:35
14-Sep-1984 13:10:04

VAX-11 BLISS-32 V4.0-742
[SMGRTL.SRC]SMGSTRTAB.B32:1

Page 21
(4)

SMG\$STRING_TABL TPARSE tables for string capabilities
 1-003 CONVERT_CONTROL - Convert ^ to control character F 10
 16-Sep-1984 01:22:35 14-Sep-1984 13:10:04 VAX-11 Bliss
 [SMGRTL.SRC]

	37	31	46	5F	59	45	4B	00402	:TPASKEYST	U.661: .ASCII	\KEY_F17\						
							FF	00409	:BYTE	-1							
							0040A	:TPASKEY\$TO	U.666: .BLKB	0							
	38	31	46	5F	59	45	4B	0040A	:TPASKEY\$T	U.668: .ASCII	\KEY_F18\						
							FF	00411	:BYTE	-1							
							00412	:TPASKEY\$TO	U.673: .BLKB	0							
	39	31	46	5F	59	45	4B	00412	:TPASKEY\$T	U.675: .ASCII	\KEY_F19\						
							FF	00419	:BYTE	-1							
							0041A	:TPASKEY\$TO	U.680: .BLKB	0							
	30	32	46	5F	59	45	4B	0041A	:TPASKEY\$T	U.682: .ASCII	\KEY_F20\						
							FF	00421	:BYTE	-1							
							00422	:TPASKEY\$TO	U.687: .BLKB	0							
57	4F	52	52	41	5F	54	46	45	4C	5F	59	45	4B	00422	:TPASKEY\$T	U.689: .ASCII	\KEY_LEFT_ARROW\
							FF	00430	:BYTE	-1							
							00431	:TPASKEY\$TO	U.694: .BLKB	0							
	53	55	4E	49	4D	5F	59	45	4B	00431	:TPASKEY\$T	U.696: .ASCII	\KEY_MINUS\				
							FF	0043A	:BYTE	-1							
							0043B	:TPASKEY\$TO	U.701: .BLKB	0							
	44	4F	49	52	45	50	5F	59	45	4B	0043B	:TPASKEY\$T	U.703: .ASCII	\KEY_PERIOD\			
							FF	00445	:BYTE	-1							
							00446	:TPASKEY\$TO	U.708: .BLKB	0							
		31	46	50	5F	59	45	4B	00446	:TPASKEY\$T	U.710: .ASCII	\KEY_PF1\					
							FF	0044D	:BYTE	-1							
							0044E	:TPASKEY\$TO	U.715: .BLKB	0							
		32	46	50	5F	59	45	4B	0044E	:TPASKEY\$T	U.717: .ASCII	\KEY_PF2\					
							FF	00455	:BYTE	-1							
							00456	:TPASKEY\$TO	U.722: .BLKB	0							
		33	46	50	5F	59	45	4B	00456	:TPASKEY\$T	U.724: .ASCII	\KEY_PF3\					
							FF	0045D	:BYTE	-1							
							0045E	:TPASKEY\$TO	U.729: .BLKB	0							
		34	46	50	5F	59	45	4B	0045E	:TPASKEY\$T	U.731: .ASCII	\KEY_PF4\					
							FF	00465	:BYTE	-1							
							00466	:TPASKEY\$TO	U.736: .BLKB	0							
							00466	:TPASKEY\$T	U.738: .ASCII	\KEY_RIGHT_ARROW\							

57	4F	52	52	41	5F	50	55	5F	59	45	48	00475	BYTE	-1	:									
												00476	:TPAS\$KEY\$TO											
												U.743:	BLKB	0										
												00476	:TPAS\$KEY\$T											
												U.745:	.ASCII	\KEY_UP_ARROW\										
												FF	00482	BYTE	-1									
												00483	:TPAS\$KEY\$TO											
												U.750:	BLKB	0										
												31	46	5F	4C	45	42	41	4C	00483	:TPAS\$KEY\$T			
												FF	00488	BYTE	-1									
												0048C	:TPAS\$KEY\$TO											
												32	46	5F	4C	45	42	41	4C	0048C	:TPAS\$KEY\$T			
												FF	00494	BYTE	-1									
												00495	:TPAS\$KEY\$TO											
												33	46	5F	4C	45	42	41	4C	00495	:TPAS\$KEY\$T			
												FF	0049D	BYTE	-1									
												0049E	:TPAS\$KEY\$TO											
												34	46	5F	4C	45	42	41	4C	0049E	:TPAS\$KEY\$T			
												FF	004A6	BYTE	-1									
												004A7	:TPAS\$KEY\$TO											
												35	46	5F	4C	45	42	41	4C	004A7	:TPAS\$KEY\$T			
												FF	004AF	BYTE	-1									
												004B0	:TPAS\$KEY\$TO											
												36	46	5F	4C	45	42	41	4C	004B0	:TPAS\$KEY\$T			
												FF	004B8	BYTE	-1									
												004B9	:TPAS\$KEY\$TO											
												37	46	5F	4C	45	42	41	4C	004B9	:TPAS\$KEY\$T			
												FF	004C1	BYTE	-1									
												004C2	:TPAS\$KEY\$TO											
												38	46	5F	4C	45	42	41	4C	004C2	:TPAS\$KEY\$T			
												FF	004CA	BYTE	-1									
												004CB	:TPAS\$KEY\$TO											
												39	46	5F	4C	45	42	41	4C	004CB	:TPAS\$KEY\$T			
												FF	004D3	BYTE	-1									
												004D4	:TPAS\$KEY\$TO											
												30	31	46	5F	4C	45	42	41	4C	004D4	:TPAS\$KEY\$T		
												FF	004DD	BYTE	-1									
												004DE	:TPAS\$KEY\$TO											

31	31	46	5F	4C	45	42	41	4C	004DE	U.820: .BLKB	0	;		
									FF 004E7	U.822: .ASCII	\LABEL_F11\	;		
									004E8	:TPASKEY\$TO	BYTE -1	;		
32	31	46	5F	4C	45	42	41	4C	004E8	U.827: .BLKB	0	;		
									FF 004F1	U.829: .ASCII	\LABEL_F12\	;		
									004F2	:TPASKEY\$TO	BYTE -1	;		
33	31	46	5F	4C	45	42	41	4C	004F2	U.834: .BLKB	0	;		
									FF 004FB	U.836: .ASCII	\LABEL_F13\	;		
									004FC	:TPASKEY\$TO	BYTE -1	;		
34	31	46	5F	4C	45	42	41	4C	004FC	U.841: .BLKB	0	;		
									FF 00505	U.843: .ASCII	\LABEL_F14\	;		
									00506	:TPASKEY\$TO	BYTE -1	;		
35	31	46	5F	4C	45	42	41	4C	00506	U.848: .BLKB	0	;		
									FF 0050F	U.850: .ASCII	\LABEL_F15\	;		
									00510	:TPASKEY\$TO	BYTE -1	;		
36	31	46	5F	4C	45	42	41	4C	00510	U.855: .BLKB	0	;		
									FF 00519	U.857: .ASCII	\LABEL_F16\	;		
									0051A	:TPASKEY\$TO	BYTE -1	;		
37	31	46	5F	4C	45	42	41	4C	0051A	U.862: .BLKB	0	;		
									FF 00523	U.864: .ASCII	\LABEL_F17\	;		
									00524	:TPASKEY\$TO	BYTE -1	;		
38	31	46	5F	4C	45	42	41	4C	00524	U.869: .BLKB	0	;		
									FF 0052D	U.871: .ASCII	\LABEL_F18\	;		
									0052E	:TPASKEY\$TO	BYTE -1	;		
39	31	46	5F	4C	45	42	41	4C	0052E	U.876: .BLKB	0	;		
									FF 00537	U.878: .ASCII	\LABEL_F19\	;		
									00538	:TPASKEY\$TO	BYTE -1	;		
30	32	46	5F	4C	45	42	41	4C	00538	U.883: .BLKB	0	;		
									FF 00541	U.885: .ASCII	\LABEL_F20\	;		
									00542	:TPASKEY\$TO	BYTE -1	;		
52	41	48	43	5F	54	5F	54	46	45	4C	00542	U.890: .BLKB	0	;
									FF 0054D	U.892: .ASCII	\LEFT_T_CHAR\	;		
									0054E	:TPASKEY\$TO	BYTE -1	;		
43	49	48	50	41	52	47	5F	46	4C	0054E	U.897: .BLKB	0	;	

4E 45 45 52 43 53 5F 54 48 47 49	FF 00558	U.899: .ASCII	\LF_GRAPHIC\	:
	00559	BYTE	-1	:
		;TPASKEY\$TO		:
		U.904: .BLKB	0	:
4E 52 4F 43 5F 54 46 45 4C 5F	00559	;TPASKEY\$T		:
52 45 57 4F 4C	FF 00565	U.906: .ASCII	\LIGHT_SCREEN\	:
	00566	BYTE	-1	:
		;TPASKEY\$TO		:
		U.911: .BLKB	0	:
4E 52 4F 43 5F 54 46 45 4C 5F	00566	;TPASKEY\$T		:
52 45 57 4F 4C	FF 00575	U.913: .ASCII	\LOWER_LEFT_CORNER\	:
	00577	BYTE	-1	:
	00578	;TPASKEY\$TO		:
		U.918: .BLKB	0	:
52 4F 43 5F 54 48 47 49 52 5F	00578	;TPASKEY\$T		:
52 45 4E 00587	U.920: .ASCII	\LOWER_RIGHT_CORNER\	:	:
	FF 0058A	BYTE	-1	:
	0058B	;TPASKEY\$TO		:
		U.925: .BLKB	0	:
52 41 48 43 5F 45 4E 49 4C 57	0058B	;TPASKEY\$T		:
45 4D 41 4E	FF 0058F	U.927: .ASCII	\NAME\	:
	00590	BYTE	-1	:
		;TPASKEY\$TO		:
		U.932: .BLKB	0	:
52 41 48 43 5F 45 4E 49 4C 57	00590	;TPASKEY\$T		:
45 4D 41 4E	FF 0059C	U.934: .ASCII	\NEWLINE_CHAR\	:
	0059D	BYTE	-1	:
		;TPASKEY\$TO		:
		U.939: .BLKB	0	:
52 41 48 43 5F 44 41 50 0059D	;TPASKEY\$T			:
	FF 005A5	U.941: .ASCII	\PAD_CHAR\	:
	005A6	BYTE	-1	:
		;TPASKEY\$TO		:
		U.946: .BLKB	0	:
5F 52 4F 53 52 55 43 5F 54 53	005A6	;TPASKEY\$T		:
45 55 51 45 52 005B5	U.948: .ASCII	\REQUEST_CURSOR_POSITION\	:	:
	FF 005BD	BYTE	-1	:
	005BE	;TPASKEY\$TO		:
		U.953: .BLKB	0	:
52 4F 53 52 55 43 5F 45 52 4F	005BE	;TPASKEY\$T		:
54 53 45 52 005CC	U.955: .ASCII	\RESTORE_CURSOR\	:	:
	005CD	BYTE	-1	:
		;TPASKEY\$TO		:
		U.960: .BLKB	0	:
58 45 44 4E 49 5F 45 53 52 45	005CD	;TPASKEY\$T		:
56 45 52 005DA	U.962: .ASCII	\REVERSE_INDEX\	:	:
	005DB	BYTE	-1	:
		;TPASKEY\$TO		:
		U.967: .BLKB	0	:
52 41 48 43 5F 54 5F 54 48 47	005DB	;TPASKEY\$T		:
49 52 005E7	U.969: .ASCII	\RIGHT_T_CHAR\	:	:
	005E8	BYTE	-1	:
		;TPASKEY\$TO		:
		U.974: .BLKB	0	:

52	4F	53	52	55	43	5F	45	56	41	53	005E8	:TPASKEYST							
										FF	005F3	U.976: .ASCII	\SAVE_CURSOR\						
											005F4	BYTE	-1						
44	52	41	57	52	4F	46	5F	4C	4C	4F	52	43	53	005F4	:TPASKEYST				
											FF	00602	U.981: .BLKB	0					
											00603	:TPASKEYST							
45	53	52	45	56	45	52	5F	4C	4C	4F	52	43	53	00603	U.983: .ASCII	\SCROLL_FORWARD\			
											FF	00611	BYTE	-1					
											00612	:TPASKEYST							
4E	45	5F	4F	54	5F	45	53	41	52	45	5F	4C	45	53	00612	U.988: .BLKB	0		
											FF	00612	U.990: .ASCII	\SCROLL_REVERSE\					
											00621	:TPASKEYST							
4E	45	5F	4F	54	5F	45	53	41	52	45	5F	4C	45	53	00621	U.995: .BLKB	0		
											FF	00621	U.997: .ASCII	\SEL_ERASE_TO_END_DISPLAY\					
											0062A	:TPASKEYST							
											0062B	U.1002: .BLKB	0						
45	4C	4F	48	57	5F	45	53	41	52	45	5F	4C	45	53	00628	:TPASKEYST			
											FF	0063A	U.1004: .ASCII	\SEL_ERASE_TO_END_LINE\					
											00640	:TPASKEYST							
											FF	00641	U.1009: .BLKB	0					
45	4C	4F	48	57	5F	45	53	41	52	45	5F	4C	45	53	00641	:TPASKEYST			
											FF	00650	U.1011: .ASCII	\SEL_ERASE_WHOLE_DISPLAY\					
											00658	:TPASKEYST							
											FF	00659	U.1016: .BLKB	0					
45	4C	4F	48	57	5F	45	53	41	52	45	5F	4C	45	53	00659	:TPASKEYST			
											FF	00668	U.1018: .ASCII	\SEL_ERASE_WHOLE_LINE\					
											0066D	:TPASKEYST							
											FF	0066E	U.1023: .BLKB	0					
4E	4F	49	54	41	43	49	4C	50	50	41	5F	54	45	53	0066E	:TPASKEYST			
											FF	0067D	U.1025: .ASCII	\SET_APPLICATION_KEYPAD\					
											00684	:TPASKEYST							
											FF	00685	U.1030: .BLKB	0					
45	53	5F	54	4F	4E	5F	52	41	48	43	5F	54	45	53	00685	:TPASKEYST			
											FF	00694	U.1032: .ASCII	\SET_CHAR_NOT_SEL_ERASE\					
											0069B	:TPASKEYST							
											FF	0069C	U.1037: .BLKB	0					
52	45	5F	4C	45	53	5F	52	41	48	43	5F	54	45	53	0069C	:TPASKEYST			
											FF	006A8	U.1039: .ASCII	\SET_CHAR_SEL_ERASE\					
											FF	006AE	:TPASKEYST						
											006AF	U.1044: .BLKB	0						

53	42	41	5F	52	4F	53	52	55	43	5F	54	45	53	006AF	TPASKEYST	U.1046: .ASCII	\SET_CURSOR_ABS\										
														FF	006BD	BYTE	-1										
														006BE	TPASKEYSTO	U.1051: .BLKB	0										
														44	41	50	006BE	TPASKEYST	U.1053: .ASCII	\SET_NUMERIC_KEYPAD\							
														FF	006CD	BYTE	-1										
														006D0	TPASKEYSTO	U.1058: .BLKB	0										
														006D1	TPASKEYST	U.1060: .ASCII	\SET_SCROLL_REGION\										
														4E	4F	006E0	TPASKEYSTO	U.1065: .BLKB	0								
														FF	006E2	TPASKEYST	U.1067: .ASCII	\SET_TAB\									
														006E3	TPASKEYSTO	U.1072: .BLKB	0										
														42	41	54	5F	54	45	53	006E3	TPASKEYST	U.1074: .ASCII	\SINGLE_HIGH\			
														FF	006EA	TPASKEYSTO	U.1079: .BLKB	0									
														006EB	TPASKEYST	U.1081: .ASCII	\TAB_CHAR\										
														FF	006F6	TPASKEYSTO	U.1086: .BLKB	0									
														006F7	TPASKEYST	U.1088: .ASCII	\TOP_T_CHAR\										
														52	41	48	43	5F	42	41	54	006F7	TPASKEYSTO	U.1093: .BLKB	0		
														FF	006FF	TPASKEYST	U.1095: .ASCII	\TRUNCATION_ICON\									
														00700	TPASKEYSTO	U.1100: .BLKB	0										
														52	41	48	43	5F	54	5F	50	4F	54	00700	TPASKEYST	U.1102: .ASCII	\UNDERLINE_CHAR\
														FF	0070A	TPASKEYSTO	U.1107: .BLKB	0									
														0070B	TPASKEYST	U.1109: .ASCII	\UPPER_LEFT_CORNER\										
														FF	0071A	TPASKEYSTO	U.1114: .BLKB	0									
														0071B	TPASKEYST	U.1116: .ASCII	\UPPER_RIGHT_CORNER\										
														52	45	00739	TPASKEYSTO	U.1118: .BLKB	0								
														FF	0073B	TPASKEYST	U.1119: .ASCII	\TOP_T_CHAR\									
														52	45	0073C	TPASKEYSTO	U.1120: .BLKB	0								
														52	45	4E	0074B	TPASKEYST	U.1122: .ASCII	\TOP_T_CHAR\							
														FF	0074E	TPASKEYST	U.1123: .BLKB	0									

52	41	42	5F	4C	41	43	49	54	52	45	56	0074F	:TPAS\$KEYSTO																
												U.1121:	.BLKB	0															
												U.1123:	.ASCII	\VERTICAL_BAR\															
												FF	0075B	.BYTE	-1														
												0075C	:TPAS\$KEYSTO																
												U.1128:	.BLKB	0															
												43	49	48	50	41	52	47	5F	54	56	0075C	:TPAS\$KEYST						
												U.1130:	.ASCII	\VT_GRAPHIC\															
												FF	00766	.BYTE	-1														
												00767	:TPAS\$KEYSTO																
												U.1135:	.BLKB	0															
												57	4F	52	52	41	4E	5F	48	54	44	49	57	00767	:TPAS\$KEYST				
												U.1137:	.ASCII	\WIDTH_NARROW\															
												FF	00773	.BYTE	-1														
												00774	:TPAS\$KEYSTO																
												U.1142:	.BLKB	0															
												45	44	49	57	5F	48	54	44	49	57	00774	:TPAS\$KEYST						
												U.1144:	.ASCII	\WIDTH_WIDE\															
												FF	0077E	.BYTE	-1														
												0077F	:TPAS\$KEYSTO																
												U.1149:	.BLKB	0															
												31	5F	52	54	53	5F	45	54	41	56	49	52	50	0077F	:TPAS\$KEYST			
												U.1151:	.ASCII	\PRIVATE_STR_1\															
												FF	0078C	.BYTE	-1														
												0078D	:TPAS\$KEYSTO																
												U.1156:	.BLKB	0															
												32	5F	52	54	53	5F	45	54	41	56	49	52	50	0078D	:TPAS\$KEYST			
												U.1158:	.ASCII	\PRIVATE_STR_2\															
												FF	0079A	.BYTE	-1														
												0079B	:TPAS\$KEYSTO																
												U.1163:	.BLKB	0															
												33	5F	52	54	53	5F	45	54	41	56	49	52	50	0079B	:TPAS\$KEYST			
												U.1165:	.ASCII	\PRIVATE_STR_3\															
												FF	007A8	.BYTE	-1														
												007A9	:TPAS\$KEYSTO																
												U.1170:	.BLKB	0															
												34	5F	52	54	53	5F	45	54	41	56	49	52	50	007A9	:TPAS\$KEYST			
												U.1172:	.ASCII	\PRIVATE_STR_4\															
												FF	007B6	.BYTE	-1														
												007B7	:TPAS\$KEYSTO																
												U.1177:	.BLKB	0															
												35	5F	52	54	53	5F	45	54	41	56	49	52	50	007B7	:TPAS\$KEYST			
												U.1179:	.ASCII	\PRIVATE_STR_5\															
												FF	007C4	.BYTE	-1														
												007C5	:TPAS\$KEYSTO																
												U.1184:	.BLKB	0															
												36	5F	52	54	53	5F	45	54	41	56	49	52	50	007C5	:TPAS\$KEYST			
												U.1186:	.ASCII	\PRIVATE_STR_6\															
												FF	007D2	.BYTE	-1														
												007D3	:TPAS\$KEYSTO																
												U.1191:	.BLKB	0															
												37	5F	52	54	53	5F	45	54	41	56	49	52	50	007D3	:TPAS\$KEYST			
												U.1193:	.ASCII	\PRIVATE_STR_7\															
												FF	007E0	.BYTE	-1														
												007E1	:TPAS\$KEYSTO																
												U.1198:	.BLKB	0															

SMG\$STRING_TABL TPARSE tables for string capabilities
1-003 CONVERT_CONTROL - Convert ^ to control

10

16-Sep-1984 01:22:35
14-Sep-1984 13:10:04

VAX-11 BLiss-32 V4.0-742
[SMGRTL.SRC]SMGSTRTAB.B32:1

Page 28
(4)

38	5F	52	54	53	SF	45	54	41	56	49	52	50	007E1	:TPASKEYST	U.1200: .ASCII	\PRIVATE_STR_8\	
												FF	007EE	.BYTE	-1		
												007EF	:TPASKEYSTO	U.1205: .BLKB	0		
39	5F	52	54	53	SF	45	54	41	56	49	52	50	007EF	:TPASKEYST	U.1207: .ASCII	\PRIVATE_STR_9\	
												FF	007FC	.BYTE	-1		
												007FD	:TPASKEYSTO	U.1212: .BLKB	0		
30	31	5F	52	54	53	SF	45	54	41	56	49	52	50	007FD	:TPASKEYST	U.1214: .ASCII	\PRIVATE_STR_10\
												FF	0080B	.BYTE	-1		
												FF	0080C	:TPASKEYFILL	U.1221: .BYTE	-1	
															.PSECT _LIB\$STATES,NOWRT, SHR, PIC,1		
													00000 SMG\$SA_STRING_STATES::				
														.BLKB	0		
													00000 BEGIN_SCAN:				
														.BLKB	0		
													99F8 00000 :TPASTYPE	U.2: .WORD	-26120		
													0000* 00002 :TPASSUBEXP	U.4: .WORD	<<U.3-U.4>-2>		
													00000000* 00004 :TPASACTION	U.5: .LONG	<<SMG\$SNEXT_RECORD-U.5>-4>		
													0000* 00008 :TPASTARGET	U.6: .WORD	<<BEGIN_SCAN-U.6>-2>		
													9021 0000A :TPASTYPE	U.7: .WORD	-28639		
													00000000* 0000C :TPASACTION	U.8: .LONG	<<SMG\$SNEXT_RECORD-U.8>-4>		
													0000* 00010 :TPASTARGET	U.9: .WORD	<<BEGIN_SCAN-U.9>-2>		
													99F8 00012 :TPASTYPE	U.10: .WORD	-26120		
													0000* 00014 :TPASSUBEXP	U.12: .WORD	<<U.11-U.12>-2>		
													00000000* 00016 :TPASACTION	U.13: .LONG	<<SMG\$BLANKS_OFF-U.13>-4>		
													0000* 0001A :TPASTARGET	U.14: .WORD	<<BEGIN_SCAN-U.14>-2>		
													15F6 0001C :TPASTYPE	U.15: .WORD	5622		
													FFFF 0001E :TPASTARGET	U.16: .WORD	-1		
													00020 :END_OF_LINE	U.3: .BLKB	0		
													11F7 00020 :TPASTYPE	U.17: .WORD	4599		
													FFFF 00022 :TPASTARGET	U.18: .WORD	-1		
													1021 00024 :TPASTYPE	U.19: .WORD	4129		
													FFFF 00026 :TPASTARGET				

N 10
16-Sep-1984 01:22:35
14-Sep-1984 13:10:04

VAX-11 Bliss-32 V4.0-742
[SMGRTL.SRC]SMGSTRTAB.B32:1

Page 29
(4)

15F6	00028	U.20: WORD	-1	:
		U.21: WORD	5622	:
FFFE	0002A	;TPA\$TARGET		:
		U.22: WORD	-2	:
	0002C	;CAPABILITY		:
		U.11: BLKB	0	:
99F8	0002C	;TPA\$TYPE		:
		U.23: WORD	-26120	:
0000*	0002E	;TPA\$SUBEXP		:
		U.25: WORD	<<U.24-U.25>-2>	:
00000000*	00030	;TPA\$ACTION		:
		U.26: LONG	<<SMG\$BLANKS_OFF-U.26>-4>	:
0000*	00034	;TPA\$TARGET		:
		U.28: WORD	<<U.27-U.28>-2>	:
1100	00036	;TPA\$TYPE		:
		U.32: WORD	4352	:
FFFE	00038	;TPA\$TARGET		:
		U.33: WORD	-2	:
1101	0003A	;TPA\$TYPE		:
		U.37: WORD	4353	:
FFFE	0003C	;TPA\$TARGET		:
		U.38: WORD	-2	:
1102	0003E	;TPA\$TYPE		:
		U.42: WORD	4354	:
FFFE	00040	;TPA\$TARGET		:
		U.43: WORD	-2	:
1103	00042	;TPA\$TYPE		:
		U.47: WORD	4355	:
0000*	00044	;TPA\$TARGET		:
		U.48: WORD	<<BEGIN_SCAN-U.48>-2>	:
9104	00046	;TPA\$TYPE		:
		U.52: WORD	-28412	:
00000000*	00048	;TPA\$ACTION		:
		U.53: LONG	<<SMG\$MISSING_END-U.53>-4>	:
FFFE	0004C	;TPA\$TARGET		:
		U.54: WORD	-2	:
9105	0004E	;TPA\$TYPE		:
		U.58: WORD	-28411	:
00000000*	00050	;TPA\$ACTION		:
		U.59: LONG	<<SMG\$MISSING_END-U.59>-4>	:
FFFE	00054	;TPA\$TARGET		:
		U.60: WORD	-2	:
85F1	00056	;TPA\$TYPE		:
		U.61: WORD	-31247	:
00000000V	00058	;TPA\$ACTION		:
		U.62: LONG	<<NOT_STRING-U.62>-4>	:
	0005C	;STRING_NAME		:
		U.24: BLKB	0	:
7106	0005C	;TPA\$TYPE		:
		U.67: WORD	28934	:
00000000*	0005E	;TPA\$ADDR		:
		U.68: LONG	<<SMG\$MASK_ADR-U.68>-4>	:
000001B9	00062	;TPA\$MASK		:
		U.69: LONG	441	:
FFFF	00066	;TPA\$TARGET		:
		U.70: WORD	-1	:

7107	00068	;TPASTYPE		
00000000*	0006A	U.74: WORD	28935	:
000001BA	0006E	;TPASADDR		:
		U.75: LONG	<<SMG\$MASK_ADR-U.75>-4>	:
FFFF	00072	;TPASMASK		:
		U.76: LONG	442	:
7108	00074	;TPASTARGET		:
		U.77: WORD	-1	:
00000000*	00076	;TPASTYPE		:
000001BB	0007A	U.81: WORD	28936	:
		;TPASADDR		:
		U.82: LONG	<<SMG\$MASK_ADR-U.82>-4>	:
FFFF	0007E	;TPASMASK		:
		U.83: LONG	443	:
7109	00080	;TPASTARGET		:
		U.84: WORD	-1	:
00000000*	00082	;TPASTYPE		:
000001BC	00086	U.88: WORD	28937	:
		;TPASADDR		:
		U.89: LONG	<<SMG\$MASK_ADR-U.89>-4>	:
FFFF	0008A	;TPASMASK		:
		U.90: LONG	444	:
710A	0008C	;TPASTARGET		:
		U.91: WORD	-1	:
00000000*	0008E	;TPASTYPE		:
000001BD	00092	U.95: WORD	28938	:
		;TPASADDR		:
		U.96: LONG	<<SMG\$MASK_ADR-U.96>-4>	:
FFFF	00096	;TPASMASK		:
		U.97: LONG	445	:
710B	00098	;TPASTARGET		:
		U.98: WORD	-1	:
00000000*	0009A	;TPASTYPE		:
000001BE	0009E	U.102: WORD	28939	:
		;TPASADDR		:
		U.103: LONG	<<SMG\$MASK_ADR-U.103>-4>	:
FFFF	000A2	;TPASMASK		:
		U.104: LONG	446	:
710C	000A4	;TPASTARGET		:
		U.105: WORD	-1	:
00000000*	000A6	;TPASTYPE		:
00000253	000AA	U.109: WORD	28940	:
		;TPASADDR		:
		U.110: LONG	<<SMG\$MASK_ADR-U.110>-4>	:
FFFF	000AE	;TPASMASK		:
		U.111: LONG	595	:
710D	000B0	;TPASTARGET		:
		U.112: WORD	-1	:
00000000*	000B2	;TPASTYPE		:
000001BF	000B6	U.116: WORD	28941	:
		;TPASADDR		:
		U.117: LONG	<<SMG\$MASK_ADR-U.117>-4>	:
FFFF	000BA	;TPASMASK		:
		U.118: LONG	447	:
710E	000BC	;TPASTARGET		:
		U.119: WORD	-1	:

00000000* 000BE	U.123: WORD	28942	:
000001C0 000C2	U.124: LONG	<<SMG\$SMASK_ADR-U.124>-4>	:
FFFF 000C6	U.125: LONG	448	:
710F 000C8	U.126: WORD	-1	:
00000000* 000CA	U.130: WORD	28943	:
000001C1 000CE	U.131: LONG	<<SMG\$SMASK_ADR-U.131>-4>	:
FFFF 000D2	U.132: LONG	449	:
7110 000D4	U.133: WORD	-1	:
00000000* 000D6	U.137: WORD	28944	:
000001C2 000DA	U.138: LONG	<<SMG\$SMASK_ADR-U.138>-4>	:
FFFF 000DE	U.139: LONG	450	:
7111 000EO	U.140: WORD	-1	:
00000000* 000E2	U.144: WORD	28945	:
00000249 000E6	U.145: LONG	<<SMG\$SMASK_ADR-U.145>-4>	:
FFFF 000EA	U.146: LONG	585	:
7112 000EC	U.147: WORD	-1	:
00000000* 000EE	U.151: WORD	28946	:
000001C3 000F2	U.152: LONG	<<SMG\$SMASK_ADR-U.152>-4>	:
FFFF 000F6	U.153: LONG	451	:
7113 000F8	U.154: WORD	-1	:
00000000* 000FA	U.158: WORD	28947	:
000001C4 000FE	U.159: LONG	<<SMG\$SMASK_ADR-U.159>-4>	:
FFFF 00102	U.160: LONG	452	:
7114 00104	U.161: WORD	-1	:
00000000* 00106	U.165: WORD	28948	:
000001C5 0010A	U.166: LONG	<<SMG\$SMASK_ADR-U.166>-4>	:
FFFF 0010E	U.167: LONG	453	:
7115 00110	U.168: WORD	-1	:
	U.172: WORD	28949	:

00000000* 00112	:TPASADDR		
0000024F 00116	;U.173: .LONG	<<SMG\$\$MASK_ADR-U.173>-4>	;
FFFF 0011A	;TPASMASK	591	;
7116 0011C	;TPASTARGET		;
7116 0011C	;U.175: .WORD	-1	;
00000000* 0011E	;U.179: .WORD	28950	;
00000247 00122	:TPASMASK		;
FFFF 00126	;U.180: .LONG	<<SMG\$\$MASK_ADR-U.180>-4>	;
7117 00128	;TPASTARGET	583	;
7117 00128	;U.182: .WORD	-1	;
00000000* 0012A	;U.186: .WORD	28951	;
00000250 0012E	:TPASADDR		;
FFFF 00132	;U.187: .LONG	<<SMG\$\$MASK_ADR-U.187>-4>	;
7118 00134	;TPASMASK	592	;
7118 00134	;U.188: .LONG		;
00000000* 00136	;U.192: .WORD	28952	;
000001C6 0013A	:TPASMASK		;
FFFF 0013E	;U.193: .LONG	<<SMG\$\$MASK_ADR-U.194>-4>	;
7119 00140	;TPASTARGET	454	;
7119 00140	;U.196: .WORD	-1	;
00000000* 00142	;U.197: .WORD	28953	;
000001C7 00146	:TPASADDR		;
FFFF 0014A	;U.201: .LONG	<<SMG\$\$MASK_ADR-U.201>-4>	;
711A 0014C	;TPASMASK	455	;
711A 0014C	;U.202: .LONG		;
00000000* 0014E	;U.207: .WORD	-1	;
000001C8 00152	:TPASADDR	28954	;
FFFF 00156	;U.208: .LONG	<<SMG\$\$MASK_ADR-U.208>-4>	;
711B 00158	;TPASMASK	456	;
711B 00158	;U.209: .LONG		;
00000000* 0015A	;U.214: .WORD	-1	;
000001C9 0015E	:TPASADDR	28955	;
FFFF 00162	;U.215: .LONG	<<SMG\$\$MASK_ADR-U.215>-4>	;
711C 00164	;TPASMASK	457	;
711C 00164	;U.216: .LONG		;
00000000* 00166	;U.217: .WORD	-1	;
00000000* 00166	;TPASADDR	28956	;

000001CA	0016A	U.222: .LONG	<<SMG\$MASK_ADR-U.222>-4>	:
FFFF	0016E	U.223: .LONG	458	:
		U.224: .WORD	-1	:
711D	00170	U.228: .WORD	28957	:
00000000*	00172	U.229: .LONG	<<SMG\$MASK_ADR-U.229>-4>	:
000001CB	00176	U.230: .LONG	459	:
FFFF	0017A	U.231: .WORD	-1	:
711E	0017C	U.235: .WORD	28958	:
00000000*	0017E	U.236: .LONG	<<SMG\$MASK_ADR-U.236>-4>	:
000001CC	00182	U.237: .LONG	460	:
FFFF	00186	U.238: .WORD	-1	:
711F	00188	U.242: .WORD	28959	:
00000000*	0018A	U.243: .LONG	<<SMG\$MASK_ADR-U.243>-4>	:
000001CD	0018E	U.244: .LONG	461	:
FFFF	00192	U.245: .WORD	-1	:
7120	00194	U.249: .WORD	28960	:
00000000*	00196	U.250: .LONG	<<SMG\$MASK_ADR-U.250>-4>	:
000001CE	0019A	U.251: .LONG	462	:
FFFF	0019E	U.252: .WORD	-1	:
7121	001A0	U.256: .WORD	28961	:
00000000*	001A2	U.257: .LONG	<<SMG\$MASK_ADR-U.257>-4>	:
000001CF	001A6	U.258: .LONG	463	:
FFFF	001AA	U.259: .WORD	-1	:
7122	001AC	U.263: .WORD	28962	:
00000000*	001AE	U.264: .LONG	<<SMG\$MASK_ADR-U.264>-4>	:
000001D0	001B2	U.265: .LONG	464	:
FFFF	001B6	U.266: .WORD	-1	:
7123	001B8	U.270: .WORD	28963	:
00000000*	001BA	U.271: .LONG	<<SMG\$MASK_ADR-U.271>-4>	:

000001D1	001BE	;TPASMASK		
FFFF	001C2	;TPAS\$TARGET	465	;
7124	001C4	;TPAS\$TYPE	-1	;
00000000*	001C6	;TPAS\$ADDR	28964	;
000001D2	001CA	;TPASMASK	<<SMG\$SMASK_ADR-U.278>-4>	;
FFFF	001CE	;TPAS\$TARGET	466	;
7125	001D0	;TPAS\$TYPE	-1	;
00000000*	001D2	;TPAS\$ADDR	28965	;
000001D3	001D6	;TPASMASK	<<SMG\$SMASK_ADR-U.285>-4>	;
FFFF	001DA	;TPAS\$TARGET	467	;
7126	001DC	;TPAS\$TYPE	-1	;
00000000*	001DE	;TPAS\$ADDR	28966	;
000001D4	001E2	;TPASMASK	<<SMG\$SMASK_ADR-U.292>-4>	;
FFFF	001E6	;TPAS\$TARGET	468	;
7127	001E8	;TPAS\$TYPE	-1	;
00000000*	001EA	;TPAS\$ADDR	28967	;
000001D5	001EE	;TPASMASK	<<SMG\$SMASK_ADR-U.299>-4>	;
FFFF	001F2	;TPAS\$TARGET	469	;
7128	001F4	;TPAS\$TYPE	-1	;
00000000*	001F6	;TPAS\$ADDR	28968	;
000001D6	001FA	;TPASMASK	<<SMG\$SMASK_ADR-U.306>-4>	;
FFFF	001FE	;TPAS\$TARGET	470	;
7129	00200	;TPAS\$TYPE	-1	;
00000000*	00202	;TPAS\$ADDR	28969	;
000001D7	00206	;TPASMASK	<<SMG\$SMASK_ADR-U.313>-4>	;
FFFF	0020A	;TPAS\$TARGET	471	;
712A	0020C	;TPAS\$TYPE	-1	;
00000000*	0020E	;TPAS\$ADDR	28970	;
000001D8	00212	;TPASMASK	<<SMG\$SMASK_ADR-U.320>-4>	;

FFFF	00216	U.321: .LONG	472	:
		U.322: .WORD	-1	:
7128	00218	U.323: .WORD	28971	:
00000000*	0021A	U.324: .WORD	<<SMG\$\$MASK_ADR-U.327>-4>	:
000001D9	0021E	U.325: .WORD	28972	:
FFFF	00222	U.326: .WORD	473	:
712C	00224	U.327: .WORD	-1	:
00000000*	00226	U.328: .WORD	28973	:
000001DA	0022A	U.329: .WORD	<<SMG\$\$MASK_ADR-U.334>-4>	:
FFFF	0022E	U.330: .WORD	474	:
712D	00230	U.331: .WORD	-1	:
00000000*	00232	U.332: .WORD	28974	:
000001DB	00236	U.333: .WORD	<<SMG\$\$MASK_ADR-U.341>-4>	:
FFFF	0023A	U.334: .WORD	475	:
712E	0023C	U.335: .WORD	-1	:
00000000*	0023E	U.336: .WORD	28975	:
0000024A	00242	U.337: .WORD	<<SMG\$\$MASK_ADR-U.348>-4>	:
FFFF	00246	U.338: .WORD	586	:
712F	00248	U.339: .WORD	-1	:
00000000*	0024A	U.340: .WORD	28976	:
000001DC	0024E	U.341: .WORD	<<SMG\$\$MASK_ADR-U.355>-4>	:
FFFF	00252	U.342: .WORD	476	:
7130	00254	U.343: .WORD	-1	:
00000000*	00256	U.344: .WORD	28977	:
000001DD	0025A	U.345: .WORD	<<SMG\$\$MASK_ADR-U.362>-4>	:
FFFF	0025E	U.346: .WORD	477	:
7131	00260	U.347: .WORD	-1	:
00000000*	00262	U.348: .WORD	28978	:
0000024C	00266	U.349: .WORD	<<SMG\$\$MASK_ADR-U.369>-4>	:
		U.350: .WORD	588	:

SMGS\$STRING_TBL TPARSE tables for string capabilities
1-003 CONVERT_CONTROL - Convert ^ to control

11

16-Sep-1984 01:22:35
14-Sep-1984 13:10:04

VAX-11 Bliss-32 V4.0-742
[SMGRTL.SRC]SMGSTRTAB.B32:1

Page 36
(4)

FFFF	0026A	:TPA\$TARGET	
7132	0026C	:TPA\$TYPE	-1
00000000*	0026E	:TPA\$ADDR	28978
00000251	00272	:TPA\$MASK	<<SMGSSMASK_ADR-U.376>-4>
FFFF	00276	:TPA\$TARGET	593
7133	00278	:TPA\$TYPE	-1
00000000*	0027A	:TPA\$ADDR	28979
000001DE	0027E	:TPA\$MASK	<<SMGSSMASK_ADR-U.383>-4>
FFFF	00282	:TPA\$TARGET	478
7134	00284	:TPA\$TYPE	-1
00000000*	00286	:TPA\$ADDR	28980
000001DF	0028A	:TPA\$MASK	<<SMGSSMASK_ADR-U.390>-4>
FFFF	0028E	:TPA\$TARGET	479
7135	00290	:TPA\$TYPE	-1
00000000*	00292	:TPA\$ADDR	28981
000001E0	00296	:TPA\$MASK	<<SMGSSMASK_ADR-U.397>-4>
FFFF	0029A	:TPA\$TARGET	480
7136	0029C	:TPA\$TYPE	-1
00000000*	0029E	:TPA\$ADDR	28982
000001E1	002A2	:TPA\$MASK	<<SMGSSMASK_ADR-U.404>-4>
FFFF	002A6	:TPA\$TARGET	481
7137	002A8	:TPA\$TYPE	-1
00000000*	002AA	:TPA\$ADDR	28983
000001E2	002AE	:TPA\$MASK	<<SMGSSMASK_ADR-U.411>-4>
FFFF	002B2	:TPA\$TARGET	482
7138	002B4	:TPA\$TYPE	-1
00000000*	002B6	:TPA\$ADDR	28984
000001E3	002BA	:TPA\$MASK	<<SMGSSMASK_ADR-U.418>-4>
FFFF	002BE	:TPA\$TARGET	483

7139	002C0	U.420: WORD	-1	:
00000000*	002C2	U.424: WORD	28985	:
000001E4	002C6	U.425: .LONG	<<SMG\$SMASK_ADR-U.425>-4>	:
FFFF	002CA	U.426: .LONG	484	:
713A	002CC	U.427: WORD	-1	:
00000000*	002CE	U.431: WORD	28986	:
000001E5	002D2	U.432: .LONG	<<SMG\$SMASK_ADR-U.432>-4>	:
FFFF	002D6	U.433: .LONG	485	:
713B	002D8	U.434: WORD	-1	:
00000000*	002DA	U.438: WORD	28987	:
000001E6	002DE	U.439: .LONG	<<SMG\$SMASK_ADR-U.439>-4>	:
FFFF	002E2	U.440: .LONG	486	:
713C	002E4	U.441: WORD	-1	:
00000000*	002E6	U.445: WORD	28988	:
000001E7	002EA	U.446: .LONG	<<SMG\$SMASK_ADR-U.446>-4>	:
FFFF	002EE	U.447: .LONG	487	:
713D	002F0	U.448: WORD	-1	:
00000000*	002F2	U.452: WORD	28989	:
000001E8	002F6	U.453: .LONG	<<SMG\$SMASK_ADR-U.453>-4>	:
FFFF	002FA	U.454: .LONG	488	:
713E	002FC	U.455: WORD	-1	:
00000000*	002FE	U.459: WORD	28990	:
000001E9	00302	U.460: .LONG	<<SMG\$SMASK_ADR-U.460>-4>	:
FFFF	00306	U.461: .LONG	489	:
713F	00308	U.462: WORD	-1	:
00000000*	0030A	U.466: WORD	28991	:
000001EA	0030E	U.467: .LONG	<<SMG\$SMASK_ADR-U.467>-4>	:
FFFF	00312	U.468: .LONG	490	:
		U.469: WORD	-1	:

7140	00314	;TPASTYPE		
00000000*	00316	;TPASADDR	28992	<<SMGSSMASK_ADR-U.474>-4>
000001EB	0031A	;TPASMASK	491	
FFFF	0031E	;TPASTARGET	-1	
7141	00320	;TPASTYPE	28993	
00000000*	00322	;TPASADDR	<<SMGSSMASK_ADR-U.481>-4>	
000001EC	00326	;TPASMASK	492	
FFFF	0032A	;TPASTARGET	-1	
7142	0032C	;TPASTYPE	28994	
00000000*	0032E	;TPASADDR	<<SMGSSMASK_ADR-U.488>-4>	
000001ED	00332	;TPASMASK	493	
FFFF	00336	;TPASTARGET	-1	
7143	00338	;TPASTYPE	28995	
000000C0*	0033A	;TPASADDR	<<SMGSSMASK_ADR-U.495>-4>	
000001EE	0033E	;TPASMASK	494	
FFFF	00342	;TPASTARGET	-1	
7144	00344	;TPASTYPE	28996	
00000000*	00346	;TPASADDR	<<SMGSSMASK_ADR-U.502>-4>	
000001EF	0034A	;TPASMASK	495	
FFFF	0034E	;TPASTARGET	-1	
7145	00350	;TPASTYPE	28997	
00000000*	00352	;TPASADDR	<<SMGSSMASK_ADR-U.509>-4>	
000001F0	00356	;TPASMASK	496	
FFFF	0035A	;TPASTARGET	-1	
7146	0035C	;TPASTYPE	28998	
00000000*	0035E	;TPASADDR	<<SMGSSMASK_ADR-U.516>-4>	
000001F1	00362	;TPASMASK	497	
FFFF	00366	;TPASTARGET	-1	
7147	00368	;TPASTYPE		

00000000*	0036A	U.522: WORD	28999	:
		U.523: LONG	<<SMG\$\$MASK_ADR-U.523>-4>	:
000001F2	0036E	U.524: LONG	498	:
FFFF	00372	U.525: WORD	-1	:
7148	00374	U.526: WORD	29000	:
00000000*	00376	U.527: WORD	29001	:
000001F3	0037A	U.528: WORD	<<SMG\$\$MASK_ADR-U.528>-4>	:
FFFF	0037E	U.529: WORD	499	:
7149	00380	U.530: WORD	-1	:
00000000*	00382	U.531: WORD	29002	:
000001F4	00386	U.532: WORD	<<SMG\$\$MASK_ADR-U.532>-4>	:
FFFF	0038A	U.533: WORD	500	:
714A	0038C	U.534: WORD	-1	:
00000000*	0038E	U.535: WORD	29003	:
000001F5	00392	U.536: WORD	<<SMG\$\$MASK_ADR-U.536>-4>	:
FFFF	00396	U.537: WORD	501	:
714B	00398	U.538: WORD	-1	:
00000000*	0039A	U.539: WORD	29004	:
000001F6	0039E	U.540: WORD	<<SMG\$\$MASK_ADR-U.540>-4>	:
FFFF	003A2	U.541: WORD	502	:
714C	003A4	U.542: WORD	-1	:
00000000*	003A6	U.543: WORD	29005	:
000001F7	003AA	U.544: WORD	<<SMG\$\$MASK_ADR-U.544>-4>	:
FFFF	003AE	U.545: WORD	503	:
714D	003B0	U.546: WORD	-1	:
00000000*	003B2	U.547: WORD	29006	:
000001F8	003B6	U.548: WORD	<<SMG\$\$MASK_ADR-U.548>-4>	:
FFFF	003BA	U.549: WORD	504	:
714E	003BC	U.550: WORD	-1	:
		U.551: WORD	29007	:
		U.552: WORD	505	:
		U.553: WORD	-1	:
		U.554: WORD	29008	:
		U.555: WORD	506	:
		U.556: WORD	-1	:
		U.557: WORD	29009	:
		U.558: WORD	507	:
		U.559: WORD	-1	:
		U.560: WORD	29010	:
		U.561: WORD	508	:
		U.562: WORD	-1	:
		U.563: WORD	29011	:
		U.564: WORD	509	:
		U.565: WORD	-1	:
		U.566: WORD	29012	:
		U.567: WORD	510	:
		U.568: WORD	-1	:
		U.569: WORD	29013	:
		U.570: WORD	511	:
		U.571: WORD	-1	:

SMGSSTRING_TBL TPARSE tables for string capabilities
1-003 CONVERT_CONTROL - Convert ^ to control

11

16-Sep-1984 01:22:35
14-Sep-1984 13:10:04

VAX-11 BLiss-32 V4.0-742
[SMGRTL.SRC]SMGSTRTAB.B32;1

Page 40
(4)

00000000*	003BE	:TPA\$ADDR	
000001F9	003C2	U.572: .LONG	<<SMG\$SMASK_ADR-U.572>-4>
FFFF	003C6	:TPA\$MASK	505
714F	003C8	U.573: .LONG	
		:TPA\$TARGET	
		U.574: .WORD	-1
00000000*	003CA	:TPA\$TYPE	
		U.578: .WORD	29007
000001FA	003CE	:TPA\$ADDR	
FFFF	003D2	U.579: .LONG	<<SMG\$SMASK_ADR-U.579>-4>
7150	003D4	:TPA\$MASK	506
		U.580: .LONG	
		:TPA\$TARGET	
		U.581: .WORD	-1
00000000*	003D6	:TPA\$TYPE	
		U.585: .WORD	29008
000001FB	003DA	:TPA\$ADDR	
FFFF	003DE	U.586: .LONG	<<SMG\$SMASK_ADR-U.586>-4>
7151	003E0	:TPA\$MASK	507
		U.587: .LONG	
		:TPA\$TARGET	
		U.588: .WORD	-1
00000000*	003E2	:TPA\$TYPE	
		U.592: .WORD	29009
000001FC	003E6	:TPA\$ADDR	
FFFF	003EA	U.593: .LONG	<<SMG\$SMASK_ADR-U.593>-4>
7152	003EC	:TPA\$MASK	508
		U.594: .LONG	
		:TPA\$TARGET	
		U.595: .WORD	-1
00000000*	003EE	:TPA\$TYPE	
		U.599: .WORD	29010
000001FD	003F2	:TPA\$ADDR	
FFFF	003F6	U.600: .LONG	<<SMG\$SMASK_ADR-U.600>-4>
7153	003F8	:TPA\$MASK	509
		U.601: .LONG	
		:TPA\$TARGET	
		U.602: .WORD	-1
00000000*	003FA	:TPA\$TYPE	
		U.606: .WORD	29011
000001FE	003FE	:TPA\$ADDR	
FFFF	00402	U.607: .LONG	<<SMG\$SMASK_ADR-U.607>-4>
7154	00404	:TPA\$MASK	510
		U.608: .LONG	
		:TPA\$TARGET	
		U.609: .WORD	-1
00000000*	00406	:TPA\$TYPE	
		U.613: .WORD	29012
000001FF	0040A	:TPA\$ADDR	
FFFF	0040E	U.614: .LONG	<<SMG\$SMASK_ADR-U.614>-4>
7155	00410	:TPA\$MASK	511
		U.615: .LONG	
		:TPA\$TARGET	
		U.616: .WORD	-1
00000000*	00412	:TPA\$TYPE	
		U.620: .WORD	29013

00000200	00416	U.621: .LONG :TPA\$MASK	<<SMG\$SMASK_ADR-U.621>-4>	:
FFFF	0041A	U.622: .LONG :TPA\$TARGET	512	:
7156	0041C	U.623: .WORD :TPA\$TYPE	-1	:
00000000*	0041E	U.627: .WORD :TPA\$ADDR	29014	:
00000201	00422	U.628: .LONG :TPA\$MASK	<<SMG\$SMASK_ADR-U.628>-4>	:
FFFF	00426	U.629: .LONG :TPA\$TARGET	513	:
7157	00428	U.630: .WORD :TPA\$TYPE	-1	:
00000000*	0042A	U.634: .WORD :TPA\$ADDR	29015	:
00000202	0042E	U.635: .LONG :TPA\$MASK	<<SMG\$SMASK_ADR-U.635>-4>	:
FFFF	00432	U.636: .LONG :TPA\$TARGET	514	:
7158	00434	U.637: .WORD :TPA\$TYPE	-1	:
00000000*	00436	U.641: .WORD :TPA\$ADDR	29016	:
00000203	0043A	U.642: .LONG :TPA\$MASK	<<SMG\$SMASK_ADR-U.642>-4>	:
FFFF	0043E	U.643: .LONG :TPA\$TARGET	515	:
7159	00440	U.644: .WORD :TPA\$TYPE	-1	:
00000000*	00442	U.648: .WORD :TPA\$ADDR	29017	:
00000204	00446	U.649: .LONG :TPA\$MASK	<<SMG\$SMASK_ADR-U.649>-4>	:
FFFF	0044A	U.650: .LONG :TPA\$TARGET	516	:
715A	0044C	U.651: .WORD :TPA\$TYPE	-1	:
00000000*	0044E	U.655: .WORD :TPA\$ADDR	29018	:
00000205	00452	U.656: .LONG :TPA\$MASK	<<SMG\$SMASK_ADR-U.656>-4>	:
FFFF	00456	U.657: .LONG :TPA\$TARGET	517	:
715B	00458	U.658: .WORD :TPA\$TYPE	-1	:
00000000*	0045A	U.662: .WORD :TPA\$ADDR	29019	:
00000206	0045E	U.663: .LONG :TPA\$MASK	<<SMG\$SMASK_ADR-U.663>-4>	:
FFFF	00462	U.664: .LONG :TPA\$TARGET	518	:
715C	00464	U.665: .WORD :TPA\$TYPE	-1	:
00000000*	00466	U.669: .WORD :TPA\$ADDR	29020	:
		U.670: .LONG :TPA\$MASK	<<SMG\$SMASK_ADR-U.670>-4>	:

00000207	0046A	:TPASMASK			
FFFF	0046E	:TPASTARGET	U.671: .LONG	519	:
715D	00470	:TPASTYPE	U.672: .WORD	-1	:
00000000*	00472	:TPASADDR	U.676: .WORD	29021	:
00000208	00476	:TPASMASK	U.677: .LONG	<<SMG\$SMASK_ADR-U.677>-4>	:
FFFF	0047A	:TPASTARGET	U.678: .LONG	520	:
715E	0047C	:TPASTYPE	U.679: .WORD	-1	:
00000000*	0047E	:TPASADDR	U.683: .WORD	29022	:
00000209	00482	:TPASMASK	U.684: .LONG	<<SMG\$SMASK_ADR-U.684>-4>	:
FFFF	00486	:TPASTARGET	U.685: .LONG	521	:
715F	00488	:TPASTYPE	U.686: .WORD	-1	:
00000000*	0048A	:TPASADDR	U.690: .WORD	29023	:
0000020A	0048E	:TPASMASK	U.691: .LONG	<<SMG\$SMASK_ADR-U.691>-4>	:
FFFF	00492	:TPASTARGET	U.692: .LONG	522	:
7160	00494	:TPASTYPE	U.693: .WORD	-1	:
00000000*	00496	:TPASADDR	U.697: .WORD	29024	:
0000020B	0049A	:TPASMASK	U.698: .LONG	<<SMG\$SMASK_ADR-U.698>-4>	:
FFFF	0049E	:TPASTARGET	U.699: .LONG	523	:
7161	004A0	:TPASTYPE	U.700: .WORD	-1	:
00000000*	004A2	:TPASADDR	U.704: .WORD	29025	:
0000020C	004A6	:TPASMASK	U.705: .LONG	<<SMG\$SMASK_ADR-U.705>-4>	:
FFFF	004AA	:TPASTARGET	U.706: .LONG	524	:
7162	004AC	:TPASTYPE	U.707: .WORD	-1	:
00000000*	004AE	:TPASADDR	U.711: .WORD	29026	:
0000020D	004B2	:TPASMASK	U.712: .LONG	<<SMG\$SMASK_ADR-U.712>-4>	:
FFFF	004B6	:TPASTARGET	U.713: .LONG	525	:
7163	004B8	:TPASTYPE	U.714: .WORD	-1	:
00000000*	004BA	:TPASADDR	U.718: .WORD	29027	:
0000020E	004BE	:TPASMASK	U.719: .LONG	<<SMG\$SMASK_ADR-U.719>-4>	:

FFFF	004C2	U.720: .LONG	526	:
7164	004C4	U.721: .WORD	-1	:
00000000*	004C6	U.725: .WORD	29028	:
0000020F	004CA	U.726: .LONG	<<SMG\$MASK_ADR-U.726>-4>	:
FFFF	004CE	U.727: .LONG	527	:
7165	004D0	U.728: .WORD	-1	:
00000000*	004D2	U.732: .WORD	29029	:
00000210	004D6	U.733: .LONG	<<SMG\$MASK_ADR-U.733>-4>	:
FFFF	004DA	U.734: .LONG	528	:
7166	004DC	U.735: .WORD	-1	:
00000000*	004DE	U.739: .WORD	29030	:
00000211	004E2	U.740: .LONG	<<SMG\$MASK_ADR-U.740>-4>	:
FFFF	004E6	U.741: .LONG	529	:
7167	004E8	U.742: .WORD	-1	:
00000000*	004EA	U.746: .WORD	29031	:
00000212	004EE	U.747: .LONG	<<SMG\$MASK_ADR-U.747>-4>	:
FFFF	004F2	U.748: .LONG	530	:
7168	004F4	U.749: .WORD	-1	:
00000000*	004F6	U.753: .WORD	29032	:
00000213	004FA	U.754: .LONG	<<SMG\$MASK_ADR-U.754>-4>	:
FFFF	004FE	U.755: .LONG	531	:
7169	00500	U.756: .WORD	-1	:
00000000*	00502	U.760: .WORD	29033	:
00000214	00506	U.761: .LONG	<<SMG\$MASK_ADR-U.761>-4>	:
FFFF	0050A	U.762: .LONG	532	:
716A	0050C	U.763: .WORD	-1	:
00000000*	0050E	U.767: .WORD	29034	:
00000215	00512	U.768: .LONG	<<SMG\$MASK_ADR-U.768>-4>	:
		U.769: .LONG	533	:

FFFF	00516	:TPASTARGET			
		U.770: WORD	-1		
716B	00518	:TPASTYPE			
		U.774: WORD	29035		
00000000*	0051A	:TPASADDR			
		U.775: .LONG	<<SMG\$MASK_ADR-U.775>-4>		
00000216	0051E	:TPASMASK			
		U.776: .LONG	534		
FFFF	00522	:TPASTARGET			
		U.777: WORD	-1		
716C	00524	:TPASTYPE			
		U.781: WORD	29036		
00000000*	00526	:TPASADDR			
		U.782: .LONG	<<SMG\$MASK_ADR-U.782>-4>		
00000217	0052A	:TPASMASK			
		U.783: .LONG	535		
FFFF	0052E	:TPASTARGET			
		U.784: WORD	-1		
716D	00530	:TPASTYPE			
		U.788: WORD	29037		
00000000*	00532	:TPASADDR			
		U.789: .LONG	<<SMG\$MASK_ADR-U.789>-4>		
00000218	00536	:TPASMASK			
		U.790: .LONG	536		
FFFF	0053A	:TPASTARGET			
		U.791: WORD	-1		
716E	0053C	:TPASTYPE			
		U.795: WORD	29038		
00000000*	0053E	:TPASADDR			
		U.796: .LONG	<<SMG\$MASK_ADR-U.796>-4>		
00000219	00542	:TPASMASK			
		U.797: .LONG	537		
FFFF	00546	:TPASTARGET			
		U.798: WORD	-1		
716F	00548	:TPASTYPE			
		U.802: WORD	29039		
00000000*	0054A	:TPASADDR			
		U.803: .LONG	<<SMG\$MASK_ADR-U.803>-4>		
0000021A	0054E	:TPASMASK			
		U.804: .LONG	538		
FFFF	00552	:TPASTARGET			
		U.805: WORD	-1		
7170	00554	:TPASTYPE			
		U.809: WORD	29040		
00000000*	00556	:TPASADDR			
		U.810: .LONG	<<SMG\$MASK_ADR-U.810>-4>		
0000021B	0055A	:TPASMASK			
		U.811: .LONG	539		
FFFF	0055E	:TPASTARGET			
		U.812: WORD	-1		
7171	00560	:TPASTYPE			
		U.816: WORD	29041		
00000000*	00562	:TPASADDR			
		U.817: .LONG	<<SMG\$MASK_ADR-U.817>-4>		
0000021C	00566	:TPASMASK			
		U.818: .LONG	540		
FFFF	0056A	:TPASTARGET			

7172 0056C	U.819: WORD	-1	:
00000000* 0056E	U.823: WORD	29042	:
0000021D 00572	U.824: LONG	<<SMG\$MASK_ADR-U.824>-4>	:
FFFF 00576	U.825: LONG	541	:
7173 00578	U.826: WORD	-1	:
00000000* 0057A	U.830: WORD	29043	:
0000021E 0057E	U.831: LONG	<<SMG\$MASK_ADR-U.831>-4>	:
FFFF 00582	U.832: LONG	542	:
7174 00584	U.833: WORD	-1	:
00000000* 00586	U.837: WORD	29044	:
0000021F 0058A	U.838: LONG	<<SMG\$MASK_ADR-U.838>-4>	:
FFFF 0058E	U.839: LONG	543	:
7175 00590	U.840: WORD	-1	:
00000000* 00592	U.844: WORD	29045	:
00000220 00596	U.845: LONG	<<SMG\$MASK_ADR-U.845>-4>	:
FFFF 0059A	U.846: LONG	544	:
7176 0059C	U.847: WORD	-1	:
00000000* 0059E	U.851: WORD	29046	:
00000221 005A2	U.852: LONG	<<SMG\$MASK_ADR-U.852>-4>	:
FFFF 005A6	U.853: LONG	545	:
7177 005A8	U.854: WORD	-1	:
00000000* 005AA	U.858: WORD	29047	:
00000222 005AE	U.859: LONG	<<SMG\$MASK_ADR-U.859>-4>	:
FFFF 005B2	U.860: LONG	546	:
7178 005B4	U.861: WORD	-1	:
00000000* 005B6	U.865: WORD	29048	:
00000223 005BA	U.866: LONG	<<SMG\$MASK_ADR-U.866>-4>	:
FFFF 005BE	U.867: LONG	547	:
	U.868: WORD	-1	:

7179	005C0	:TPA\$TYPE		
00000000*	005C2	U.872: WORD	29049	
00000224	005C6	:TPA\$ADDR		<<SMG\$MASK_ADR-U.873>-4>
FFFF	005CA	U.873: .LONG		
		:TPA\$MASK		
		U.874: .LONG	548	
		:TPA\$TARGET		
717A	005CC	U.875: WORD	-1	
00000000*	005CE	:TPA\$TYPE		29050
00000225	005D2	U.879: WORD		<<SMG\$MASK_ADR-U.880>-4>
FFFF	005D6	:TPA\$ADDR		
		U.880: .LONG		
		:TPA\$MASK		
		U.881: .LONG	549	
		:TPA\$TARGET		
717B	005D8	U.882: WORD	-1	
00000000*	005DA	:TPA\$TYPE		29051
00000226	005DE	U.886: WORD		<<SMG\$MASK_ADR-U.887>-4>
FFFF	005E2	:TPA\$ADDR		
		U.887: .LONG		
		:TPA\$MASK		
		U.888: .LONG	550	
		:TPA\$TARGET		
717C	005E4	U.889: WORD	-1	
00000000*	005E6	:TPA\$TYPE		29052
00000227	005EA	U.893: WORD		<<SMG\$MASK_ADR-U.894>-4>
FFFF	005EE	:TPA\$ADDR		
		U.894: .LONG		
		:TPA\$MASK		
		U.895: .LONG	551	
		:TPA\$TARGET		
717D	005F0	U.896: WORD	-1	
00000000*	005F2	:TPA\$TYPE		29053
0000024B	005F6	U.900: WORD		<<SMG\$MASK_ADR-U.901>-4>
FFFF	005FA	:TPA\$ADDR		
		U.901: .LONG		
		:TPA\$MASK		
		U.902: .LONG	587	
		:TPA\$TARGET		
717E	005FC	U.903: WORD	-1	
00000000*	005FE	:TPA\$TYPE		29054
00000228	00602	U.907: WORD		<<SMG\$MASK_ADR-U.908>-4>
FFFF	00606	:TPA\$ADDR		
		U.908: .LONG		
		:TPA\$MASK		
		U.909: .LONG	552	
		:TPA\$TARGET		
717F	00608	U.910: WORD	-1	
00000000*	0060A	:TPA\$TYPE		29055
00000229	0060E	U.914: WORD		<<SMG\$MASK_ADR-U.915>-4>
FFFF	00612	:TPA\$ADDR		
		U.915: .LONG		
		:TPA\$MASK		
		U.916: .LONG	553	
		:TPA\$TARGET		
7180	00614	U.917: WORD	-1	

00000000* 00616	U.921: WORD	29056	:
0000022A 0061A	U.922: LONG	<<SMG\$MASK_ADR-U.922>-4>	:
FFFF 0061E	U.923: LONG	554	:
7181 00620	U.924: WORD	-1	:
00000000* 00622	U.928: WORD	29057	:
0000022B 00626	U.929: LONG	<<SMG\$MASK_ADR-U.929>-4>	:
FFFF 0062A	U.930: LONG	555	:
7182 0062C	U.931: WORD	-1	:
00000000* 0062E	U.935: WORD	29058	:
0000022C 00632	U.936: LONG	<<SMG\$MASK_ADR-U.936>-4>	:
FFFF 00636	U.937: LONG	556	:
7183 00638	U.938: WORD	-1	:
00000000* 0063A	U.942: WORD	29059	:
0000022D 0063E	U.943: LONG	<<SMG\$MASK_ADR-U.943>-4>	:
FFFF 00642	U.944: LONG	557	:
7184 00644	U.945: WORD	-1	:
00000000* 00646	U.949: WORD	29060	:
00000248 0064A	U.950: LONG	<<SMG\$MASK_ADR-U.950>-4>	:
FFFF 0064E	U.951: LONG	584	:
7185 00650	U.952: WORD	-1	:
00000000* 00652	U.956: WORD	29061	:
0000022E 00656	U.957: LONG	<<SMG\$MASK_ADR-U.957>-4>	:
FFFF 0065A	U.958: LONG	558	:
7186 0065C	U.959: WORD	-1	:
00000000* 0065E	U.963: WORD	29062	:
00000252 00662	U.964: LONG	<<SMG\$MASK_ADR-U.964>-4>	:
FFFF 00666	U.965: LONG	594	:
7187 00668	U.966: WORD	-1	:
	U.970: WORD	29063	:

00000000*	0066A	:TPASADDR	U.971: .LONG	<<SMG\$SMASK_ADR-U.971>-4>
0000022F	0066E	:TPASMASK	U.972: .LONG	559
FFFF	00672	:TPASTARGET	U.973: .WORD	-1
7188	00674	:TPASTYPE	U.977: .WORD	29064
00000000*	00676	:TPASADDR	U.978: .LONG	<<SMG\$SMASK_ADR-U.978>-4>
00000230	0067A	:TPASMASK	U.979: .LONG	560
FFFF	0067E	:TPASTARGET	U.980: .WORD	-1
7189	00680	:TPASTYPE	U.984: .WORD	29065
00000000*	00682	:TPASADDR	U.985: .LONG	<<SMG\$SMASK_ADR-U.985>-4>
00000231	00686	:TPASMASK	U.986: .LONG	561
FFFF	0068A	:TPASTARGET	U.987: .WORD	-1
718A	0068C	:TPASTYPE	U.991: .WORD	29066
00000000*	0068E	:TPASADDR	U.992: .LONG	<<SMG\$SMASK_ADR-U.992>-4>
00000232	00692	:TPASMASK	U.993: .LONG	562
FFFF	00696	:TPASTARGET	U.994: .WORD	-1
718B	00698	:TPASTYPE	U.998: .WORD	29067
00000000*	0069A	:TPASADDR	U.999: .LONG	<<SMG\$SMASK_ADR-U.999>-4>
00000233	0069E	:TPASMASK	U.1000: .LONG	563
FFFF	006A2	:TPASTARGET	U.1001: .WORD	-1
718C	006A4	:TPASTYPE	U.1005: .WORD	29068
00000000*	006A6	:TPASADDR	U.1006: .LONG	<<SMG\$SMASK_ADR-U.1006>-4>
00000234	006AA	:TPASMASK	U.1007: .LONG	564
FFFF	006AE	:TPASTARGET	U.1008: .WORD	-1
718D	006B0	:TPASTYPE	U.1012: .WORD	29069
00000000*	006B2	:TPASADDR	U.1013: .LONG	<<SMG\$SMASK_ADR-U.1013>-4>
00000235	006B6	:TPASMASK	U.1014: .LONG	565
FFFF	006BA	:TPASTARGET	U.1015: .WORD	-1
718E	006BC	:TPASTYPE	U.1019: .WORD	29070
00000000*	006BE	:TPASADDR		

00000236	006C2	U.1020: .LONG ;TPASMASK	566	<<SMG\$SMASK_ADR-U.1020>-4>
FFFF	006C6	U.1021: .LONG ;TPASTARGET	-1	;
718F	006C8	U.1022: .WORD ;TPASTYPE	;	
00000000*	006CA	U.1026: .WORD ;TPASADDR	29071	;
00000237	006CE	U.1027: .LONG ;TPASMASK	567	;
FFFF	006D2	U.1028: .LONG ;TPASTARGET	-1	;
7190	006D4	U.1029: .WORD ;TPASTYPE	;	
00000000*	006D6	U.1033: .WORD ;TPASADDR	29072	;
00000238	006DA	U.1034: .LONG ;TPASMASK	568	;
FFFF	006DE	U.1035: .LONG ;TPASTARGET	-1	;
7191	006E0	U.1036: .WORD ;TPASTYPE	;	
00000000*	006E2	U.1040: .WORD ;TPASADDR	29073	;
00000239	006E6	U.1041: .LONG ;TPASMASK	569	;
FFFF	006EA	U.1042: .LONG ;TPASTARGET	-1	;
7192	006EC	U.1043: .WORD ;TPASTYPE	;	
00000000*	006EE	U.1047: .WORD ;TPASADDR	29074	;
0000023A	006F2	U.1048: .LONG ;TPASMASK	570	;
FFFF	006F6	U.1049: .LONG ;TPASTARGET	-1	;
7193	006F8	U.1050: .WORD ;TPASTYPE	;	
00000000*	006FA	U.1054: .WORD ;TPASADDR	29075	;
0000023B	006FE	U.1055: .LONG ;TPASMASK	571	;
FFFF	00702	U.1056: .LONG ;TPASTARGET	-1	;
7194	00704	U.1057: .WORD ;TPASTYPE	;	
00000000*	00706	U.1061: .WORD ;TPASADDR	29076	;
0000023C	0070A	U.1062: .LONG ;TPASMASK	572	;
FFFF	0070E	U.1063: .LONG ;TPASTARGET	-1	;
7195	00710	U.1064: .WORD ;TPASTYPE	;	
00000000*	00712	U.1068: .WORD ;TPASADDR	29077	;
		U.1069: .LONG ;<<SMG\$SMASK_ADR-U.1069>-4>		;

SMG\$STRING_TABL TPARSE tables for string capabilities
1-003 CONVERT_CONTROL - Convert ^ to control

112

16-Sep-1984 01:22:35
14-Sep-1984 13:10:04

VAX-11 Blfs-32 V4.0-742
[SMGRTL.SRC]SMGSTRTAB.B32:1

Page 50
(4)

57
1-

0000023D	00716	:TPA\$MASK	
	FFFF	:TPASTARGET	573
7196	0071C	:TPASTYPE	-1
00000000*	0071E	:TPASADDR	29078
0000023E	00722	:TPASMASK	<<SMG\$SMASK_ADR-U.1076>-4>
	FFFF	:TPASTARGET	574
7197	00728	:TPASTYPE	-1
00000000*	0072A	:TPASADDR	29079
0000023F	0072E	:TPASMASK	<<SMG\$SMASK_ADR-U.1083>-4>
	FFFF	:TPASTARGET	575
7198	00734	:TPASTYPE	-1
00000000*	00736	:TPASADDR	29080
00000240	0073A	:TPASMASK	<<SMG\$SMASK_ADR-U.1090>-4>
	FFFF	:TPASTARGET	576
7199	00740	:TPASTYPE	-1
00000000*	00742	:TPASADDR	29081
0000024E	00746	:TPASMASK	<<SMG\$SMASK_ADR-U.1097>-4>
	FFFF	:TPASTARGET	590
719A	0074C	:TPASTYPE	-1
00000000*	0074E	:TPASADDR	29082
00000241	00752	:TPASMASK	<<SMG\$SMASK_ADR-U.1104>-4>
	FFFF	:TPASTARGET	577
719B	00758	:TPASTYPE	-1
00000000*	0075A	:TPASADDR	29083
00000242	0075E	:TPASMASK	<<SMG\$SMASK_ADR-U.1111>-4>
	FFFF	:TPASTARGET	578
719C	00764	:TPASTYPE	-1
00000000*	00766	:TPASADDR	29084
00000243	0076A	:TPASMASK	<<SMG\$SMASK_ADR-U.1118>-4>

FFFF	0076E	U.1119: .LONG	579	:
719D	00770	U.1120: .WORD	-1	:
00000000*	00772	U.1124: .WORD	29085	:
00000244	00776	U.1125: .LONG	<<SMG\$MASK_ADR-U.1125>-4>	:
FFFF	0077A	U.1126: .LONG	580	:
719E	0077C	U.1127: .WORD	-1	:
00000000*	0077E	U.1131: .WORD	29086	:
0000024D	00782	U.1132: .LONG	<<SMG\$MASK_ADR-U.1132>-4>	:
FFFF	00786	U.1133: .LONG	589	:
719F	00788	U.1134: .WORD	-1	:
00000000*	0078A	U.1138: .WORD	29087	:
00000245	0078E	U.1139: .LONG	<<SMG\$MASK_ADR-U.1139>-4>	:
FFFF	00792	U.1140: .LONG	581	:
71A0	00794	U.1141: .WORD	-1	:
00000000*	00796	U.1145: .WORD	29088	:
00000246	0079A	U.1146: .LONG	<<SMG\$MASK_ADR-U.1146>-4>	:
FFFF	0079E	U.1147: .LONG	582	:
71A1	007A0	U.1148: .WORD	-1	:
00000000*	007A2	U.1152: .WORD	29089	:
0000028B	007A6	U.1153: .LONG	<<SMG\$MASK_ADR-U.1153>-4>	:
FFFF	007AA	U.1154: .LONG	651	:
71A2	007AC	U.1155: .WORD	-1	:
00000000*	007AE	U.1159: .WORD	29090	:
0000028C	007B2	U.1160: .LONG	<<SMG\$MASK_ADR-U.1160>-4>	:
FFFF	007B6	U.1161: .LONG	652	:
71A3	007B8	U.1162: .WORD	-1	:
00000000*	007BA	U.1166: .WORD	29091	:
0000028D	007BE	U.1167: .LONG	<<SMG\$MASK_ADR-U.1167>-4>	:
		U.1168: .LONG	653	:

SMG\$STRING_TABL TPARSE tables for string capabilities
1-003 CONVERT_CONTROL - Convert ^ to control

• K 12

16-Sep-1984 01:22:35
14-Sep-1984 13:10:04

VAX-11 Bliss-32 V4.0-742
[SMGRTL.SRC]SMGSTRTAB.B32:1

Page 52
(4)

FFFF	007C2	;TPA\$TARGET	
71A4	007C4	U.1169: WORD	-1
00000000*	007C6	;TPA\$TYPE	29092
0000028E	007CA	U.1173: WORD	<<SMG\$SMASK_ADR-U.1174>-4>
FFFF	007CE	;TPA\$ADDR	
		U.1174: LONG	
		;TPA\$MASK	
		U.1175: LONG	654
71A5	007D0	;TPA\$TARGET	
		U.1176: WORD	-1
00000000*	007D2	;TPA\$TYPE	29093
0000028F	007D6	U.1180: WORD	<<SMG\$SMASK_ADR-U.1181>-4>
FFFF	007DA	;TPA\$ADDR	
		U.1181: LONG	
		;TPA\$MASK	
		U.1182: LONG	655
71A6	007DC	;TPA\$TARGET	
		U.1183: WORD	-1
00000000*	007DE	;TPA\$TYPE	29094
00000290	007E2	U.1187: WORD	<<SMG\$SMASK_ADR-U.1188>-4>
FFFF	J07E6	;TPA\$ADDR	
		U.1188: LONG	
		;TPA\$MASK	
		U.1189: LONG	656
71A7	007E8	;TPA\$TARGET	
		U.1190: WORD	-1
00000000*	007EA	;TPA\$TYPE	29095
00000291	007EE	U.1194: WORD	<<SMG\$SMASK_ADR-U.1195>-4>
FFFF	007F2	;TPA\$ADDR	
		U.1195: LONG	
		;TPA\$MASK	
		U.1196: LONG	657
71A8	007F4	;TPA\$TARGET	
		U.1197: WORD	-1
00000000*	007F6	;TPA\$TYPE	29096
00000292	007FA	U.1201: WORD	<<SMG\$SMASK_ADR-U.1202>-4>
FFFF	007FE	;TPA\$ADDR	
		U.1202: LONG	
		;TPA\$MASK	
		U.1203: LONG	658
71A9	00800	;TPA\$TARGET	
		U.1204: WORD	-1
00000000*	00802	;TPA\$TYPE	29097
00000293	00806	U.1208: WORD	<<SMG\$SMASK_ADR-U.1209>-4>
FFFF	0080A	;TPA\$ADDR	
		U.1209: LONG	
		;TPA\$MASK	
		U.1210: LONG	659
71AA	0080C	;TPA\$TARGET	
		U.1211: WORD	-1
00000000*	0080E	;TPA\$TYPE	29098
00000294	00812	U.1215: WORD	<<SMG\$SMASK_ADR-U.1216>-4>
FFFF	00816	;TPA\$ADDR	
		U.1216: LONG	
		;TPA\$MASK	
		U.1217: LONG	660
		;TPA\$TARGET	

15F6	00818	U.1218: WORD	-1	:
FFFE	0081A	U.1219: WORD	5622	:
		U.1220: WORD	-2	:
	0081C	U.EQUALS_STRING		:
		U.27: BLKB	0	:
99F8	0081C	U.TPASTYPE		:
		U.1222: WORD	-26120	:
0000*	0081E	U.TPAS\$SUBEXP		:
00000000*	00820	U.1223: WORD	<<U.3-U.1223>-2>	:
0000*	00824	U.TPAS\$TARGET	<<SMG\$NEXT_RECORD-U.1224>-4>	:
903D	00826	U.1225: WORD	<<U.27-U.1225>-2>	:
00000000*	00828	U.TPAS\$ACTION	-28611	:
0000*	0082C	U.1226: WORD	<<SMG\$STORE_CAP_MASK-U.1227>-4>	:
81F1	0082E	U.TPASTYPE	<<U.1228-U.1229>-2>	:
00000000*	00830	U.1227: LONG	-32271	:
85ED	00834	U.1229: WORD	<<SMG\$SYNTAX_ERROR-U.1231>-4>	:
00000000*	00836	U.TPAS\$ACTION	-31251	:
		U.1230: WORD	<<SMG\$SYNTAX_ERROR-U.1233>-4>	:
	0083A	U.STRING_CAP_VALUE		:
99F8	0083A	U.1231: BLRB	0	:
0000*	0083C	U.TPAS\$SUBEXP	-26120	:
00000000*	0083E	U.1232: WORD	<<U.3-U.1235>-2>	:
0000*	00842	U.TPAS\$TARGET	<<SMG\$NEXT_RECORD-U.1236>-4>	:
19F8	00844	U.1233: WORD	<<U.1228-U.1237>-2>	:
0000*	00846	U.TPAS\$SUBEXP	6648	:
0000*	00848	U.1234: WORD	<<U.1239-U.1240>-2>	:
81F1	0084A	U.TPASTYPE	<<BEGIN_SCAN-U.1241>-2>	:
00000000*	0084C	U.1235: WORD	-32271	:
85ED	00850	U.1236: LONG	<<SMG\$SYNTAX_ERROR-U.1243>-4>	:
00000000*	00852	U.TPAS\$ACTION	-31251	:
		U.1237: WORD	<<SMG\$SYNTAX_ERROR-U.1245>-4>	:
902C	00856	U.COMMA: BLKB	0	:
00000000*	00858	U.TPAS\$ACTION	-28628	:

FFFF 0085C	U.1247: .LONG	<<SMG\$\$BLANKS_OFF-U.1247>-4>	:
	U.1248: .WORD	-1	:
91F2 0085E	U.1249: .WORD	-28174	:
00000000* 00860	U.1250: .LONG	<<SMG\$\$BLANKS_OFF-U.1250>-4>	:
0000* 00864	U.1251: .WORD	<<COMMA-U.1251>-2>	:
99F8 00866	U.1252: .WORD	-26120	:
0000* 00868	U.1253: .WORD	<<U.3-U.1253>-2>	:
00000000* 0086A	U.1254: .LONG	<<SMG\$\$NEXT_RECORD-U.1254>-4>	:
FFFF 0086E	U.1255: .WORD	-1	:
85ED 00870	U.1256: .WORD	-31251	:
00000000* 00872	U.1257: .LONG	<<SMG\$\$SYNTAX_ERROR-U.1257>-4>	:
	00876: STRING_CAP	0	:
9027 00876	U.1259: .BLKB	-28633	:
00000000* 00878	U.1258: .WORD	<<SMG\$\$BLANKS_ON-U.1259>-4>	:
0000* 0087C	U.1259: .LONG	<<SMG\$\$BLANKS_ON-U.1263>-4>	:
9022 0087E	U.1260: .WORD	<<U.1260-U.1261>-2>	:
00000000* 00880	U.1262: .WORD	-28638	:
0000* 00884	U.1263: .LONG	<<SMG\$\$BLANKS_ON-U.1263>-4>	:
15ED 00886	U.1264: .WORD	<<U.1264-U.1265>-2>	:
FFFE 00888	U.1265: .WORD	5613	:
	U.1266: .WORD	-2	:
	0088A: SINGLE_QUOTE_STRING	0	:
19F8 0088A	U.1267: .WORD	6648	:
0000* 0088C	U.1268: .WORD	<<U.1269-U.1270>-2>	:
0000* 0088E	U.1269: .WORD	<<U.1260-U.1271>-2>	:
9024 00890	U.1270: .WORD	-28636	:
00000000V 00892	U.1271: .WORD	<<CONVERT_ESCAPE-U.1273>-4>	:
0000* 00896	U.1272: .WORD	<<U.1260-U.1274>-2>	:
19F8 00898	U.1273: .WORD	6648	:
0000* 0089A	U.1274: .WORD	<<U.1276-U.1277>-2>	:

0000* 0089C	;TPASTARGET		
19F8 0089E	U.1278: WORD	<<U.1260-U.1278>-2>	:
0000* 008A0	;TPASSUBEXP	6648	:
0000* 008A2	;TPASTARGET	<<U.1280-U.1281>-2>	:
19F8 008A4	;TPASTYPE	<<U.1260-U.1282>-2>	:
0000* 008A6	U.1283: WORD	6648	:
0000* 008A8	;TPASTARGET	<<U.1284-U.1285>-2>	:
9027 008AA	U.1286: WORD	<<U.1260-U.1286>-2>	:
	;TPASTYPE	-28633	:
00000000* 008AC	;TPASACTION	<<SMG\$COPY_CAP-U.1288>-4>	:
0000* 008B0	;TPASTARGET	<<COMMA-U.1289>-2>	:
91F2 008B2	;TPASTYPE	-28174	:
00000000* 008B4	;TPASACTION	<<SMG\$SAVE_TOKEN_STRING-U.1291>-4>	:
0000* 008B8	;TPASTARGET	<<U.1260-U.1292>-2>	:
8022 008BA	;TPASTYPE	-32734	:
00000000* 008BC	;TPASACTION	<<SMG\$SYNTAX_ERROR-U.1294>-4>	:
91ED 008C0	;TPASTYPE	-28179	:
00000000* 008C2	;TPASACTION	<<SMG\$SAVE_TOKEN_STRING-U.1296>-4>	:
0000* 008C6	;TPASTARGET	<<U.1260-U.1297>-2>	:
15F6 008C8	;TPASTYPE	-1	:
FFFF 008CA	;TPASTARGET	5622	:
	U.1299: WORD	-1	:
008CC	;DOUBLE_QUOTE_STRING	0	:
19F8 008CC	;TPASTYPE		:
	U.1300: WORD	6648	:
0000* 008CE	;TPASSUBEXP	<<U.1269-U.1301>-2>	:
0000* 008D0	;TPASTARGET	<<U.1264-U.1302>-2>	:
9024 008D2	;TPASTYPE	-28636	:
00000000V 008D4	;TPASACTION	<<CONVERT_ESCAPE-U.1304>-4>	:
0000* 008D8	;TPASTARGET	<<U.1264-U.1305>-2>	:
19F8 008DA	;TPASTYPE	6648	:
0000* 008DC	;TPASSUBEXP		:

0000* 008DE	U.1307: WORD	<<U.1276-U.1307>-2>	:
19F8 008E0	U.1308: WORD	<<U.1264-U.1308>-2>	:
0000* 008E2	U.1309: WORD	6648	:
0000* 008E4	U.1310: WORD	<<U.1280-U.1310>-2>	:
19F8 008E6	U.1311: WORD	<<U.1264-U.1311>-2>	:
0000* 008E8	U.1312: WORD	6648	:
0000* 008EA	U.1313: WORD	<<U.1284-U.1313>-2>	:
9022 008EC	U.1314: WORD	<<U.1264-U.1314>-2>	:
00000000* 008EE	U.1315: WORD	-28638	:
0000* 008F2	U.1316: LONG	<<SMG\$COPY_CAP-U.1316>-4>	:
91F2 008F4	U.1317: WORD	<<COMMA-U.1317>-2>	:
00000000* 008F6	U.1318: WORD	-28174	:
0000* 008FA	U.1319: LONG	<<SMG\$SAVE_TOKEN_STRING-U.1319>-4>	:
8027 008FC	U.1320: WORD	<<U.1264-U.1320>-2>	:
00000000* 008FE	U.1321: WORD	-32729	:
91ED 00902	U.1322: LONG	<<SMG\$SYNTAX_ERROR-U.1322>-4>	:
00000000* 00904	U.1323: WORD	-28179	:
0000* 00908	U.1324: LONG	<<SMG\$SAVE_TOKEN_STRING-U.1324>-4>	:
15F6 0090A	U.1325: WORD	<<U.1264-U.1325>-2>	:
FFFF 0090C	U.1326: WORD	5622	:
0090E	U.1327: WORD	-1	:
105F 0090E	U.1269: BLKB	0	:
0000* 00910	U.1328: WORD	4191	:
15F6 00912	U.1330: WORD	<<U.1329-U.1330>-2>	:
FFFF 00914	U.1331: WORD	5622	:
00916	U.1332: WORD	-2	:
9024 00916	U.1329: BLKB	0	:
00000000V 00918	U.1333: WORD	-28636	:
	U.1334: LONG	<<INSERT_DOLLAR-U.1334>-4>	:

FFFF	0091C	:TPA\$TARGET		
905E	0091E	:TPA\$TYPE	U.1335: WORD	-1
00000000V	00920	:TPA\$ACTION	U.1336: WORD	-28578
FFFF	00924	:TPA\$TARGET	U.1337: .LONG	<<INSERT_CARROT-U.1337>-4>
9021	00926	:TPA\$TYPE	U.1338: WORD	-1
00000000V	00928	:TPA\$ACTION	U.1339: WORD	-28639
FFFF	0092C	:TPA\$TARGET	U.1340: .LONG	<<INSERT_EXCLAMATION-U.1340>-4>
9028	0092E	:TPA\$TYPE	U.1341: WORD	-1
00000000V	00930	:TPA\$ACTION	U.1342: WORD	-28632
FFFF	00934	:TPA\$TARGET	U.1343: .LONG	<<INSERT_PARENTHESIS-U.1343>-4>
15F6	00936	:TPA\$TYPE	U.1344: WORD	-1
FFFE	00938	:TPA\$TARGET	U.1345: WORD	5622
0093A		:CARROT	U.1346: WORD	-2
105E	0093A	:TPA\$TYPE	U.1276: BLKB	0
0000*	0093C	:TPA\$TARGET	U.1347: WORD	4190
15F6	0093E	:TPA\$TYPE	U.1349: WORD	<<U.1348-U.1349>-2>
FFFE	00940	:TPA\$TARGET	U.1350: WORD	5622
00942		:CONTROL_CHAR	U.1351: WORD	-2
95ED	00942	:TPA\$TYPE	U.1348: BLKB	0
00000000*	00944	:TPA\$ACTION	U.1352: WORD	-27155
FFFF	00948	:TPA\$TARGET	U.1353: .LONG	<<CONVERT_CONTROL-U.1353>-4>
0094A		:FAO STRING	U.1354: WORD	-1
8021	0094A	:TPA\$TYPE	U.1280: BLKB	0
00000000*	0094C	:TPA\$ACTION	U.1355: WORD	-32735
15F6	00950	:TPA\$TYPE	U.1356: .LONG	<<SMG\$SAVE_TOKEN_STRING-U.1356>-4>
FFFE	00952	:TPA\$TARGET	U.1357: WORD	5622
8055	00954	:TPA\$TYPE	U.1358: WORD	-2
00000000*	00956	:TPA\$ACTION	U.1359: WORD	-32683
91F1	0095A	:TPA\$TYPE	U.1360: .LONG	<<SMG\$SAVE_TOKEN_STRING-U.1360>-4>

00000000V 0095C	U.1361: WORD	-28175	:
FFE 00960	:TPASACTION		:
	U.1362: .LONG	<<INVALID_DIRECTIVE-U.1362>-4>	:
	:TPASTARGET		:
	U.1363: WORD	-2	:
95ED 00962	:TPASTYPE		:
	U.1364: WORD	-27155	:
00000000V 00964	:TPASACTION		:
FFE 00968	U.1365: .LONG	<<INVALID_DIRECTIVE-U.1365>-4>	:
	:TPASTARGET		:
	U.1366: WORD	-2	:
F04C 0096A	:TPASTYPE		:
	U.1367: WORD	-4020	:
00000000* 0096C	:TPASACTION		:
	U.1368: .LONG	<<SMG\$\$\$SAVE_TOKEN_STRING-U.1368>-4>	:
00000000* 00970	:TPASADDR		:
	U.1369: .LONG	<<SMG\$\$MASK_ADR-U.1369>-4>	:
FFFFFF 00974	:TPASMASK		:
	U.1370: .LONG	-1	:
FFFF 00978	:TPASTARGET		:
	U.1371: WORD	-1	:
91F1 0097A	:TPASTYPE		:
	U.1372: WORD	-28175	:
00000000V 0097C	:TPASACTION		:
FFE 00980	U.1373: .LONG	<<INVALID_DIRECTIVE-U.1373>-4>	:
	:TPASTARGET		:
	U.1374: WORD	-2	:
95ED 00982	:TPASTYPE		:
	U.1375: WORD	-27155	:
00000000V 00984	:TPASACTION		:
FFE 00988	U.1376: .LONG	<<INVALID_DIRECTIVE-U.1376>-4>	:
	:TPASTARGET		:
	U.1377: WORD	-2	:
0098A	:ARITH_CAP		:
	U.1284: BLKB	0	:
9028 0098A	:TPASTYPE		:
	U.1378: WORD	-28632	:
00000000V 0098C	:TPASACTION		:
	U.1379: .LONG	<<INIT_ARITH_CAP-U.1379>-4>	:
0000* 00990	:TPASTARGET		:
	U.1381: WORD	<<U.1380-U.1381>-2>	:
11F2 00992	:TPASTYPE		:
	U.1382: WORD	4594	:
0000* 00994	:TPASTARGET		:
	U.1383: WORD	<<U.1284-U.1383>-2>	:
15F6 00996	:TPASTYPE		:
	U.1384: WORD	5622	:
FFE 00998	:TPASTARGET		:
	U.1385: WORD	-2	:
0099A	:ARITH_CAP2		:
	U.1380: BLKB	0	:
19F8 0099A	:TPASTYPE		:
	U.1386: WORD	6648	:
0000* 0099C	:TPASSUBEXP		:
	U.1388: WORD	<<U.1387-U.1388>-2>	:
FFFF 0099E	:TPASTARGET		:
	U.1389: WORD	-1	:

15F6	009A0	:TPASTYPE	
FFFE	009A2	:TPASTARGET	5622
	009A4	U.1391: .WORD	-2
		OPERATOR: BLKB	0
702B	009A4	U.1390: .WORD	28715
00000000*	009A6	:TPASADDR	<<SMG\$\$MASK_ADR-U.1393>-4>
FFFFFFFB	009AA	:TPASMASK	
	0000*	U.1394: .LONG	-5
	009AE	:TPASTARGET	<<U.1387-U.1395>-2>
702D	009B0	U.1395: .WORD	28717
00000000*	009B2	:TPASADDR	<<SMG\$\$MASK_ADR-U.1397>-4>
FFFFFFFA	009B6	:TPASMASK	
	0000*	U.1397: .LONG	-6
	009BA	:TPASTARGET	<<U.1387-U.1399>-2>
702A	009BC	U.1399: .WORD	28714
00000000*	009BE	:TPASADDR	<<SMG\$\$MASK_ADR-U.1401>-4>
FFFFFFF9	009C2	:TPASMASK	
	0000*	U.1401: .LONG	-7
	009C6	:TPASTARGET	<<U.1387-U.1403>-2>
702F	009C8	U.1403: .WORD	28719
00000000*	009CA	:TPASADDR	<<SMG\$\$MASK_ADR-U.1405>-4>
FFFFFFF8	009CE	:TPASMASK	
	0000*	U.1405: .LONG	-8
	009D2	:TPASTARGET	<<U.1387-U.1407>-2>
9029	009D4	U.1407: .WORD	-28631
00000000V	009D6	:TPASACTION	<<END_ARITH_CAP-U.1409>-4>
FFFF	009DA	:TPASTARGET	
85ED	009DC	U.1410: .WORD	-1
00000000V	009DE	:TPASACTION	-31251
	009E2	U.1411: .WORD	<<EXPRESSION_ERROR-U.1412>-4>
91F3	009E2	:TPASTYPE	0
00000000V	009E4	U.1412: .LONG	-28173
	0000*	OPERAND	<<STORE_OPERAND-U.1414>-4>
	009E8	U.1387: .BLKB	
19F8	009EA	U.1413: .WORD	<<OPERATOR-U.1415>-2>
		:TPASTARGET	
		U.1414: .LONG	
		:TPASTARGET	
		U.1415: .WORD	
		:TPASTYPE	

0000* 009EC	U.1416: .WORD	6648	:
	;TPASSUBEXP		:
0000* 009EE	U.1418: .WORD	<<U.1417-U.1418>-2>	:
	;TPASTARGET		:
81F1 009F0	U.1419: .WORD	<<OPERATOR-U.1419>-2>	:
	;TPASTYPE		:
00000000V 009F2	U.1420: .WORD	-32271	:
85ED 009F6	U.1421: .LONG	<<EXPRESSION_ERROR-U.1421>-4>	:
	;TPASTYPE		:
00000000V 009F8	U.1422: .WORD	-31251	:
	;TPASACTION		:
009FC	U.1423: .LONG	<<EXPRESSION_ERROR-U.1423>-4>	:
	;SUBSTITUTION		:
0025 009FC	U.1417: .BLKB	0	:
	;TPASTYPE		:
15F6 009FE	U.1424: .WORD	37	:
	;TPASTARGET		:
FFFE 00A00	U.1425: .WORD	5622	:
	;TPASTARGET		:
91F3 00A02	U.1426: .WORD	-2	:
	;TPASTYPE		:
00000000V 00A04	U.1427: .WORD	-28173	:
	;TPASACTION		:
FFFF 00A08	U.1428: .LONG	<<STORE_SUBSTITUTION-U.1428>-4>	:
	;TPASTARGET		:
85ED 00A0A	U.1429: .WORD	-1	:
	;TPASTYPE		:
00000000V 00A0C	U.1430: .WORD	-31251	:
	;TPASACTION		:
	U.1431: .LONG	<<EXPRESSION_ERROR-U.1431>-4>	:

.PSECT _LIB\$KEYOS,NOWRT, SHR, PIC,1

00000	SMG\$SA_STRING_KEYWDS::		:	
	BLKB	0	:	
00000	;TPASKEY0		:	
0000*	00000	U.1: .BLKB	0	:
0000*	00002	;TPASKEY		:
0000*	00004	U.30: .WORD	<U.29-U.1>	:
0000*	00006	U.35: .WORD	<U.34-U.1>	:
0000*	00008	U.40: .WORD	<U.39-U.1>	:
0000*	00010	U.45: .WORD	<U.44-U.1>	:
0000*	00012	U.50: .WORD	<U.49-U.1>	:
0000*	00014	U.56: .WORD	<U.55-U.1>	:
0000*	00016	U.65: .WORD	<U.64-U.1>	:
0000*	00018	U.72: .WORD	<U.71-U.1>	:
0000*	00020	U.79: .WORD	<U.78-U.1>	:

0000* 00014	U.86: .WORD	<U.85-U.1>
	U.93: .WORD	<U.92-U.1>
0000* 00016	U.100: .WORD	<U.99-U.1>
	U.107: .WORD	<U.106-U.1>
0000* 00018	U.114: .WORD	<U.113-U.1>
	U.121: .WORD	<U.120-U.1>
0000* 0001C	U.128: .WORD	<U.127-U.1>
	U.135: .WORD	<U.134-U.1>
0000* 00020	U.142: .WORD	<U.141-U.1>
	U.149: .WORD	<U.148-U.1>
0000* 00022	U.156: .WORD	<U.155-U.1>
	U.163: .WORD	<U.162-U.1>
0000* 00024	U.170: .WORD	<U.169-U.1>
	U.177: .WORD	<U.176-U.1>
0000* 0002E	U.184: .WORD	<U.183-U.1>
	U.191: .WORD	<U.190-U.1>
0000* 00030	U.198: .WORD	<U.197-U.1>
	U.205: .WORD	<U.204-U.1>
0000* 00032	U.212: .WORD	<U.211-U.1>
	U.219: .WORD	<U.218-U.1>
0000* 0003A	U.226: .WORD	<U.225-U.1>
	U.233: .WORD	<U.232-U.1>
0000* 0003C	U.240: .WORD	<U.239-U.1>
	U.247: .WORD	<U.246-U.1>
0000* 00040	U.254: .WORD	<U.253-U.1>
	U.261: .WORD	<U.260-U.1>
0000* 00042	U.268: .WORD	<U.267-U.1>
	U.275: .WORD	<U.274-U.1>
0000* 00044	U.282: .WORD	<U.281-U.1>

0000* 0004C	:TPASKEY		
0000* 0004E	U.289: .WORD	<U.288-U.1>	
0000* 00050	U.296: .WORD	<U.295-U.1>	
0000* 00052	U.303: .WORD	<U.302-U.1>	
0000* 00054	U.310: .WORD	<U.309-U.1>	
0000* 00056	U.317: .WORD	<U.316-U.1>	
0000* 00058	U.324: .WORD	<U.323-U.1>	
0000* 0005A	U.331: .WORD	<U.330-U.1>	
0000* 0005C	U.338: .WORD	<U.337-U.1>	
0000* 0005E	U.345: .WORD	<U.344-U.1>	
0000* 00060	U.352: .WORD	<U.351-U.1>	
0000* 00062	U.359: .WORD	<U.358-U.1>	
0000* 00064	U.366: .WORD	<U.365-U.1>	
0000* 00066	U.373: .WORD	<U.372-U.1>	
0000* 00068	U.380: .WORD	<U.379-U.1>	
0000* 0006A	U.387: .WORD	<U.386-U.1>	
0000* 0006C	U.394: .WORD	<U.393-U.1>	
0000* 0006E	U.401: .WORD	<U.400-U.1>	
0000* 00070	U.408: .WORD	<U.407-U.1>	
0000* 00072	U.415: .WORD	<U.414-U.1>	
0000* 00074	U.422: .WORD	<U.421-U.1>	
0000* 00076	U.429: .WORD	<U.428-U.1>	
0000* 00078	U.436: .WORD	<U.435-U.1>	
0000* 0007A	U.443: .WORD	<U.442-U.1>	
0000* 0007C	U.450: .WORD	<U.449-U.1>	
0000* 0007E	U.457: .WORD	<U.456-U.1>	
0000* 00080	U.464: .WORD	<U.463-U.1>	
0000* 00082	U.471: .WORD	<U.470-U.1>	
0000* 00084	U.478: .WORD	<U.477-U.1>	

0000* 00086	U.485: .WORD	<U.484-U.1>
0000* 00088	U.492: .WORD	<U.491-U.1>
0000* 0008A	U.499: .WORD	<U.498-U.1>
0000* 0008C	U.506: .WORD	<U.505-U.1>
0000* 0008E	U.513: .WORD	<U.512-U.1>
0000* 00090	U.520: .WORD	<U.519-U.1>
0000* 00092	U.527: .WORD	<U.526-U.1>
0000* 00094	U.534: .WORD	<U.533-U.1>
0000* 00096	U.541: .WORD	<U.540-U.1>
0000* 00098	U.548: .WORD	<U.547-U.1>
0000* 0009A	U.555: .WORD	<U.554-U.1>
0000* 0009C	U.562: .WORD	<U.561-U.1>
0000* 0009E	U.569: .WORD	<U.568-U.1>
0000* 000A0	U.576: .WORD	<U.575-U.1>
0000* 000A2	U.583: .WORD	<U.582-U.1>
0000* 000A4	U.590: .WORD	<U.589-U.1>
0000* 000A6	U.597: .WORD	<U.596-U.1>
0000* 000A8	U.604: .WORD	<U.603-U.1>
0000* 000AA	U.611: .WORD	<U.610-U.1>
0000* 000AC	U.618: .WORD	<U.617-U.1>
0000* 000AE	U.625: .WORD	<U.624-U.1>
0000* 000B0	U.632: .WORD	<U.631-U.1>
0000* 000B2	U.639: .WORD	<U.638-U.1>
0000* 000B4	U.646: .WORD	<U.645-U.1>
0000* 000B6	U.653: .WORD	<U.652-U.1>
0000* 000B8	U.660: .WORD	<U.659-U.1>
0000* 000BA	U.667: .WORD	<U.666-U.1>
0000* 000BC	U.674: .WORD	<U.673-U.1>
	U.681: .WORD	<U.680-U.1>

0000* 000BE	:TPASKEY		
0000* 000C0	U.688: .WORD	<U.687-U.1>	;
0000* 000C2	U.695: .WORD	<U.694-U.1>	;
0000* 000C4	U.702: .WORD	<U.701-U.1>	;
0000* 000C6	U.709: .WORD	<U.708-U.1>	;
0000* 000C8	U.716: .WORD	<U.715-U.1>	;
0000* 000CA	U.723: .WORD	<U.722-U.1>	;
0000* 000CC	U.730: .WORD	<U.729-U.1>	;
0000* 000CE	U.737: .WORD	<U.736-U.1>	;
0000* 000D0	U.744: .WORD	<U.743-U.1>	;
0000* 000D2	U.751: .WORD	<U.750-U.1>	;
0000* 000D4	U.758: .WORD	<U.757-U.1>	;
0000* 000D6	U.765: .WORD	<U.764-U.1>	;
0000* 000D8	U.772: .WORD	<U.771-U.1>	;
0000* 000DA	U.779: .WORD	<U.778-U.1>	;
0000* 000DC	U.786: .WORD	<U.785-U.1>	;
0000* 000DE	U.793: .WORD	<U.792-U.1>	;
0000* 000E0	U.800: .WORD	<U.799-U.1>	;
0000* 000E2	U.807: .WORD	<U.806-U.1>	;
0000* 000E4	U.814: .WORD	<U.813-U.1>	;
0000* 000E6	U.821: .WORD	<U.820-U.1>	;
0000* 000E8	U.828: .WORD	<U.827-U.1>	;
0000* 000EA	U.835: .WORD	<U.834-U.1>	;
0000* 000EC	U.842: .WORD	<U.841-U.1>	;
0000* 000EE	U.849: .WORD	<U.848-U.1>	;
0000* 000F0	U.856: .WORD	<U.855-U.1>	;
0000* 000F2	U.863: .WORD	<U.862-U.1>	;
0000* 000F4	U.870: .WORD	<U.869-U.1>	;
0000* 000F6	U.877: .WORD	<U.876-U.1>	;

0000* 000F8	U.884: .WORD	<U.883-U.1>	:
0000* 000FA	U.891: .WORD	<U.890-U.1>	:
0000* 000FC	U.898: .WORD	<U.897-U.1>	:
0000* 000FE	U.905: .WORD	<U.904-U.1>	:
0000* 00100	U.912: .WORD	<U.911-U.1>	:
0000* 00102	U.919: .WORD	<U.918-U.1>	:
0000* 00104	U.926: .WORD	<U.925-U.1>	:
0000* 00106	U.933: .WORD	<U.932-U.1>	:
0000* 00108	U.940: .WORD	<U.939-U.1>	:
0000* 0010A	U.947: .WORD	<U.946-U.1>	:
0000* 0010C	U.954: .WORD	<U.953-U.1>	:
0000* 0010E	U.961: .WORD	<U.960-U.1>	:
0000* 00110	U.968: .WORD	<U.967-U.1>	:
0000* 00112	U.975: .WORD	<U.974-U.1>	:
0000* 00114	U.982: .WORD	<U.981-U.1>	:
0000* 00116	U.989: .WORD	<U.988-U.1>	:
0000* 00118	U.996: .WORD	<U.995-U.1>	:
0000* 0011A	U.1003: .WORD	<U.1002-U.1>	:
0000* 0011C	U.1010: .WORD	<U.1009-U.1>	:
0000* 0011E	U.1017: .WORD	<U.1016-U.1>	:
0000* 00120	U.1024: .WORD	<U.1023-U.1>	:
0000* 00122	U.1031: .WORD	<U.1030-U.1>	:
0000* 00124	U.1038: .WORD	<U.1037-U.1>	:
0000* 00126	U.1045: .WORD	<U.1044-U.1>	:
0000* 00128	U.1052: .WORD	<U.1051-U.1>	:
0000* 0012A	U.1059: .WORD	<U.1058-U.1>	:
0000* 0012C	U.1066: .WORD	<U.1065-U.1>	:
0000* 0012E	U.1073: .WORD	<U.1072-U.1>	:
	U.1080: .WORD	<U.1079-U.1>	:

0000* 00130	;TPASKEY	
0000* 00132	U.1087: .WORD	<U.1086-U.1>
0000* 00134	;TPASKEY	
0000* 00136	U.1094: .WORD	<U.1093-U.1>
0000* 00138	;TPASKEY	
0000* 0013A	U.1101: .WORD	<U.1100-U.1>
0000* 0013C	;TPASKEY	
0000* 0013E	U.1108: .WORD	<U.1107-U.1>
0000* 0013A	;TPASKEY	
0000* 0013C	U.1115: .WORD	<U.1114-U.1>
0000* 0013E	;TPASKEY	
0000* 00140	U.1122: .WORD	<U.1121-U.1>
0000* 00142	;TPASKEY	
0000* 00144	U.1129: .WORD	<U.1128-U.1>
0000* 00146	;TPASKEY	
0000* 00148	U.1136: .WORD	<U.1135-U.1>
0000* 0014A	;TPASKEY	
0000* 0014C	U.1143: .WORD	<U.1142-U.1>
0000* 0014E	;TPASKEY	
0000* 00150	U.1150: .WORD	<U.1149-U.1>
0000* 00152	;TPASKEY	
0000* 00154	U.1157: .WORD	<U.1156-U.1>
	;TPASKEY	
	U.1164: .WORD	<U.1163-U.1>
	;TPASKEY	
	U.1171: .WORD	<U.1170-U.1>
	;TPASKEY	
	U.1178: .WORD	<U.1177-U.1>
	;TPASKEY	
	U.1185: .WORD	<U.1184-U.1>
	;TPASKEY	
	U.1192: .WORD	<U.1191-U.1>
	;TPASKEY	
	U.1199: .WORD	<U.1198-U.1>
	;TPASKEY	
	U.1206: .WORD	<U.1205-U.1>
	;TPASKEY	
	U.1213: .WORD	<U.1212-U.1>

.EXTRN	OTSSCVT TI L, SMG\$BLANKS_OFF
.EXTRN	SMG\$BLANKS_ON, SMG\$COPY_CAP
.EXTRN	SMG\$FLUSH_ARITHMETIC
.EXTRN	SMG\$FLUSH_SAVED_BUFFER
.EXTRN	SMG\$MISSING_END
.EXTRN	SMG\$NEXT_RECORD
.EXTRN	SMG\$SAVE_TOKEN_STRING
.EXTRN	SMG\$STORE_CAP_MASK
.EXTRN	SMG\$SYNTAX_ERROR
.EXTRN	SMG\$ERRAT [IN, SMG\$INVDIR
.EXTRN	SMG\$INVEXP, SMG\$MISTERNAM
.EXTRN	SMG\$NOTSTRCAP, SMG\$SYNERR
.EXTRN	SMG\$MASK_ADR, SMG\$NEXT_NEGATIVE_NUMBER
.EXTRN	SMG\$CURRENT_LINE
.EXTRN	SMG\$DATA_OFFSET

.PSECT _SMG\$CODE,NOWRT, SHR, PIC,2

SMG\$STRING_TABL TPARSE tables for string capabilities
1-003 CONVERT_CONTROL - Convert ^ to control

113

16-Sep-1984 01:22:35
14-Sep-1984 13:10:04

VAX-11 Bliss-32 V4.0-742
[SMGRTL.SRC]SMGSTRTAB.B32;1

Page 67
(4)

: Routine Size: 57 bytes, Routine Base: _SMG\$CODE + 0000

625 0698 1 XSBTTL 'CONVERT_ESCAPE - Convert \$ to escape character'
626 0699 1 ROUTINE CONVERT_ESCAPE =
627 0700 1
628 0701 1 ++
629 0702 1 FUNCTIONAL DESCRIPTION:
630 0703 1
631 0704 1 Stores an escape character in the current capability string in
632 0705 1 TERMTABLE.EXE.
633 0706 1
634 0707 1 CALLING SEQUENCE:
635 0708 1
636 0709 1 status = CONVERT_ESCAPE ()
637 0710 1
638 0711 1 FORMAL PARAMETERS:
639 0712 1
640 0713 1 NONE
641 0714 1
642 0715 1 IMPLICIT INPUTS:
643 0716 1
644 0717 1 AP Points to TPARSE parameter block
645 0718 1
646 0719 1 IMPLICIT OUTPUTS:
647 0720 1
648 0721 1 NONE
649 0722 1
650 0723 1 COMPLETION STATUS:
651 0724 1
652 0725 1 SSS_NORMAL
653 0726 1
654 0727 1 SIDE EFFECTS:
655 0728 1
656 0729 1
657 0730 1 --
658 0731 2 BEGIN
659 0732 2 BUILTIN
660 0733 2 CALLG,
661 0734 2 AP;
662 0735 2 MAP
663 0736 2 AP : REF BLOCK [,BYTE];
664 0737 2 LITERAL
665 0738 2 K_ESCAPE = %X'1B';
666 0739 2 BIND
667 0740 2 CAP_PTRS = .AP [PARAM_L_CUR_TERM_DEF] : VECTOR [,WORD];
668 0741 2
669 0742 2
670 0743 2
671 0744 2
672 0745 2
673 0746 2
674 0747 2
675 0748 2
676 0749 2
677 0750 2
678 0751 2
679 0752 2
680 0753 2
681 0754 2
+ The NAME capability should have preceded this. Complain if it didn't.
-
+ IF CAP_PTRS EQL 0
THEN
SIGNAL_STOP (SMGS_MISTERNAM);
+ Move an escape character into the capability string instead of
the \$.
Part of the string may already be copied - append in this part so

```

: 682      0755 2 ! as not to overwrite it.
: 683      0756 2 !-
: 684      0757 2
: 685      0758 2 CALLG (.AP, SMG$FLUSH_SAVED_BUFFER);
: 686      0759 2
: 687      0760 2 !+
: 688      0761 2 ! Insert the escape character.
: 689      0762 2 !-
: 690      0763 2
: 691      0764 2 AP [PARAM_L_SAVED_TOKENCNT] = 1;
: 692      0765 2 AP [PARAM_L_SAVED_TOKENSTR] = UPLIT (BYTE (K_ESCAPE));
: 693      0766 2 ! escape now 'saved'
: 694      0767 2 CALLG (.AP, SMG$FLUSH_SAVED_BUFFER);
: 695      0768 2
: 696      0769 2 RETURN (SS$_NORMAL);
: 697      0770 2
: 698      0771 1 END:                                ! end of routine CONVERT_ESCAPE

```

```

00039      .BLKB   3
1B 0003C P.AAA: .BYTE 27

```

0004 00000 CONVERT_ESCAPE:							
52	00000000G	00	9E	00002	WORD	Save R2	0699
48		AC	D5	00009	MOVAB	SMG\$FLUSH_SAVED_BUFFER, R2	
		0D	12	0000C	TSTL	72(AP)	0746
00000000G	00	0000000G	00	9F	BNEQ	1\$	
54	00		01	FB	PUSHAB	SMG\$MISTERNAM	0748
58	62		01	FA	CALLS	#1, [IBSTOP	
	AC		0001B	1\$:	CALLG	(AP) SMG\$FLUSH_SAVED_BUFFER	0758
	DA		01	D0	MOVL	#1, 84(AP)	0764
	50		AF	00022	MOVAB	P.AAA, 88(AP)	0765
	62		6C	FA	CALLG	(AP), SMG\$FLUSH_SAVED_BUFFER	0767
	01		01	D0	MOVL	#1, R0	0769
	04			0002A	RET		0771

; Routine Size: 46 bytes. Routine Base: _SMG\$CODE + 003D

```

700 0772 1 XSBTTL 'EXPRESSION_ERROR - Signal and expression error'
701 0773 1 ROUTINE EXPRESSION_ERROR =
702 0774 1
703 0775 1 ++
704 0776 1 FUNCTIONAL DESCRIPTION:
705 0777 1
706 0778 1 Signal that an invalid arithmetic expression was found.
707 0779 1
708 0780 1 CALLING SEQUENCE:
709 0781 1
710 0782 1
711 0783 1 status = EXPRESSION_ERROR ()
712 0784 1
713 0785 1 FORMAL PARAMETERS:
714 0786 1
715 0787 1 NONE
716 0788 1
717 0789 1 IMPLICIT INPUTS:
718 0790 1
719 0791 1 AP Points to TPARSE parameter block
720 0792 1
721 0793 1 IMPLICIT OUTPUTS:
722 0794 1
723 0795 1 NONE
724 0796 1
725 0797 1 COMPLETION STATUS:
726 0798 1
727 0799 1 SSS_NORMAL
728 0800 1
729 0801 1 SIDE EFFECTS:
730 0802 1
731 0803 1 --
732 0804 1
733 0805 2 BEGIN
734 0806 2 BUILTIN
735 0807 2 AP;
736 0808 2 MAP
737 0809 2 AP : REF BLOCK [,BYTE];
738 0810 2
739 0811 2 SIGNAL_STOP (SMGS_ERRAT LIN,
740 0812 2 3, .SMG$SCURRENT_LINE,
741 0813 2 .AP [TPASL_TOKENCNT],
742 0814 2 .AP [TPASL_TOKENPTR],
743 0815 2 SMGS_INVEXP)
744 0816 2
745 0817 1 END: ! end of routine EXPRESSION_ERROR

```

0000 00000 EXPRESSION_ERROR:

7E	0000000G	00	9F	00002	WORD	Save nothing
	10	AC	7D	00008	PUSHAB	SMGS_INVEXP
	0000000G	00	DD	0000C	MOVO	16(AP), -(SP)
	03	DD	00012	PUSHL	SMG\$SCURRENT_LINE	
				PUSHL	#3	

```

: 0773
: 0811
: 0813
: 0812
: 0811

```

SMG\$STRING_TBL TPARSE tables for string capabilities
1-003 EXPRESSION_ERROR - Signal and expression error

D 14

16-Sep-1984 01:22:35

14-Sep-1984 13:10:04

VAX-11 Bliss-32 V4.0-742
[SMGRTL.SRC]SMGSTRTAB.B32:1

Page 71
(6)

00000000G 00 00000000G 00 9F 00014
00000000G 00 06 FB 0001A
04 00021

PUSHAB SMG\$ERRAT LIN
CALLS #6, [IB\$STOP
RET

; 0817

: Routine Size: 34 bytes. Routine Base: _SMGSCODE + 006B

747 0818 1 ZSBTTL 'END_ARITH_CAP - End of arithmetic capability data'
748 0819 1 ROUTINE END_ARITH_CAP =
749 0820 1
750 0821 1 !++
751 0822 1 FUNCTIONAL DESCRIPTION:
752 0823 1
753 0824 1 We have found the end of an arithmetic expression (a right
754 0825 1 parenthesis). An arithmetic expression should be terminated
755 0826 1 by the encoded type SMG\$K_STORE, indicating that the final
756 0827 1 value has been computed.
757 0828 1
758 0829 1 CALLING SEQUENCE:
759 0830 1
760 0831 1 status = END_ARITH_CAP ()
761 0832 1
762 0833 1 FORMAL PARAMETERS:
763 0834 1
764 0835 1 NONE
765 0836 1
766 0837 1 IMPLICIT INPUTS:
767 0838 1
768 0839 1 AP Points to TPARSE parameter block
769 0840 1
770 0841 1 IMPLICIT OUTPUTS:
771 0842 1
772 0843 1 NONE
773 0844 1
774 0845 1 COMPLETION STATUS:
775 0846 1
776 0847 1 SSS_NORMAL
777 0848 1
778 0849 1 SIDE EFFECTS:
779 0850 1
780 0851 1 --
781 0852 1
782 0853 2 BEGIN
783 0854 2
784 0855 2 BUILTIN
785 0856 2 CALLG,
786 0857 2 AP;
787 0858 2 MAP
788 0859 2 AP : REF BLOCK [,BYTE];
789 0860 2
790 0861 2
791 0862 2 !+ Indicate end of computing.
792 0863 2 !-
793 0864 2
794 0865 2 AP [PARAM_L_SAVED_TOKENCNT] = 1;
795 0866 2 AP [PARAM_L_SAVED_TOKENSTR] = UPLIT (BYTE (SMG\$K_STORE));
796 0867 2
797 0868 2 CALLG (.AP, SMG\$\$FLUSH_ARITHMETIC);
798 0869 2 ! move the data
799 0870 2
800 0871 2 AP [TPASV_BLANKS] = 1; ! look at blanks again
801 0872 2
802 0873 2 RETURN (SSS_NORMAL);
803 0874 2

SMG\$STRING_TABL TPARSE tables for string capabilities
1-003 END_ARITH_CAP - End of arithmetic capability da

F 14
16-Sep-1984 01:22:35 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 13:10:04 [SMGRTL.SRC]SMGSTRTAB.B32;1

Page 73
(7)

: 804 0875 1 END;

! end of routine END_ARITH_CAP

F7 0008D .BLKB 3
F7 00090 P.AAB: .BYTE -9

0000 00000 END_ARITH_CAP:
0000000G 00 01 D0 00002 .WORD Save nothing : 0819
54 AC AF 9E 00006 MOVL #1, 84(AP) : 0865
58 AC 6C FA 0000B MOVAB P.AAB, 88(AP) : 0866
04 AC 01 88 00012 CALLG (AP) { SMG\$FLUSH_ARITHMETIC : 0868
50 01 D0 00016 BISB2 #1, 4(AP) : 0871
04 00019 MOVL #1, R0 : 0873
RET : 0875

: Routine Size: 26 bytes. Routine Base: _SMGSCODE + 0091

```

806 0876 1 XSBTTL 'INIT_ARITH_CAP - Initialize arithmetic capability data'
807 0877 1 ROUTINE INIT_ARITH_CAP =
808 0878 1
809 0879 1 !++
810 0880 1 FUNCTIONAL DESCRIPTION:
811 0881 1
812 0882 1 We have found the beginning of an arithmetic expression (a left
813 0883 1 parenthesis). Before moving operands and operators into the data
814 0884 1 area, initialize it as an arithmetic capability, ie. start with a
815 0885 1 negative length and a type of SMG$K_ARITH_STRING.
816 0886 1
817 0887 1 CALLING SEQUENCE:
818 0888 1
819 0889 1     status = INIT_ARITH_CAP ()
820 0890 1
821 0891 1 FORMAL PARAMETERS:
822 0892 1
823 0893 1     NONE
824 0894 1
825 0895 1 IMPLICIT INPUTS:
826 0896 1
827 0897 1     AP      Points to TPARSE parameter block
828 0898 1
829 0899 1 IMPLICIT OUTPUTS:
830 0900 1
831 0901 1     NONE
832 0902 1
833 0903 1 COMPLETION STATUS:
834 0904 1
835 0905 1     SSS_NORMAL
836 0906 1
837 0907 1 SIDE EFFECTS:
838 0908 1
839 0909 1 --
840 0910 1
841 0911 2 BEGIN
842 0912 2
843 0913 2 BUILTIN
844 0914 2     CALLG,
845 0915 2     AP;
846 0916 2     MAP
847 0917 2     AP : REF BLOCK [,BYTE];
848 0918 2
849 0919 2 LOCAL
850 0920 2     FIRST : INITIAL (0),           ! flag to indicate start of string
851 0921 2     START_CAP_STRING;          ! where to copy data
852 0922 2     MAP
853 0923 2     START_CAP_STRING : REF VECTOR [,BYTE,SIGNED];
854 0924 2
855 0925 2
856 0926 2     BIND
857 0927 2     CAP_PTRS = .AP [PARAM_L_CUR_TERM_DEF] : VECTOR [,WORD];
858 0928 2
859 0929 2     !+ The NAME capability should have preceded any arithmetic capability.
860 0930 2     Complain if it didn't.
861 0931 2
862 0932 2

```

```
863      0933 2      IF CAP_PTRS EQL 0
864      0934 2      THEN
865      0935 2      SIGNAL_STOP (SMG$MISTERNAM);
866      0936 2
867      0937 2
868      0938 2      + We may have copied some normal text before we encountered an
869      0939 2      arithmetic expression. Find the start of the string, and make
870      0940 2      sure that there isn't some more saved text.
871      0941 2
872      0942 2
873      0943 2      CALLG (.AP, SMG$FLUSH_SAVED_BUFFER);
874      0944 2
875      0945 2      IF .CAP_PTRS [.AP [PARAM_L_CUR_CAP_NUMBER]] NEQ 0
876      0946 2      THEN
877      0947 2      BEGIN
878      0948 2      + Not start of the capability string.
879      0949 2
880      0950 2
881      0951 2      START_CAP_STRING = .AP [PARAM_L_CUR_TERM_DEF] +
882      0952 2      :CAP_PTRS [.AP [PARAM_L_CUR_CAP_NUMBER]];
883      0953 2
884      0954 2      ELSE
885      0955 2      + First byte of capability string. Set offset pointer.
886      0956 2
887      0957 2      BEGIN
888      0958 2
889      0959 2      FIRST = 1;
890      0960 2      START_CAP_STRING = .AP [PARAM_L_CUR_DATA_BYTE];
891      0961 2      CAP_PTRS [.AP [PARAM_L_CUR_CAP_NUMBER]] = .SMG$DATA_OFFSET;
892      0962 2      SMG$DATA_OFFSET = .SMG$DATA_OFFSET + 2;
893      0963 2      + leave space for at least size, type
894      0964 2      : bytes
895      0965 2
896      0966 2
897      0967 2      IF .FIRST
898      0968 2      THEN
899      0969 2      AP [PARAM_L_CUR_DATA_BYTE] = .AP [PARAM_L_CUR_DATA_BYTE] + 2;
900      0970 2      + space for size & type bytes
901      0971 2
902      0972 2
903      0973 2      + If some other things preceeded this arithmetic expression, we need
904      0974 2      to move them over by 2 bytes to make room for a size and type.
905      0975 2
906      0976 2      IF NOT .FIRST
907      0977 2      THEN
908      0978 2      BEGIN
909      0979 2      LOCAL
910      0980 2      TYPE:
911      0981 2      TYPE = .START_CAP_STRING [1]; ! make this a longword
912      0982 2
913      0983 2      IF .TYPE NEQ SMG$K_ARITH_STRING
914      0984 2      THEN
915      0985 4      BEGIN
916      0986 4      + ! first expression
917      0987 4      + We don't allow mixing of FAO strings and arithmetic
918      0988 4      expressions. We know the type isn't arithmetic, so
919      0989 4      ! if the length is negative, there must be an FAO string.
```

```

920      0990 4
921      0991 4
922      0992 4
923      0993 4
924      0994 4
925      0995 4
926      0996 4
927      0997 4
928      0998 4
929      0999 4
930      1000 4
931      1001 4
932      1002 4
933      1003 4
934      1004 4
935      1005 4
936      1006 4
937      1007 4
938      1008 4
939      1009 4
940      1010 4
941      1011 4
942      1012 4
943      1013 4
944      1014 4
945      1015 4
946      1016 4
947      1017 4
948      1018 4
949      1019 4
950      1020 4
951      1021 4
952      1022 3
953      1023 3
954      1024 3
955      1025 2
956      1026 2
957      1027 2
958      1028 2
959      1029 2
960      1030 2
961      1031 2
962      1032 2
963      1033 2
964      1034 2
965      1035 2
966      1036 2
967      1037 2
968      1038 1

      !-
      IF .START_CAP_STRING [0] LSS 0
      THEN
          SIGNAL_STOP (SMGS$ERRAT LIN,
                      3, .SMGS$CURRENT LINE,
                      .AP [PARAM_L_SAVED_TOKENCNT],
                      .AP [PARAM_L_SAVED_TOKENSTR],
                      SMGS$SYNERR);

      !+
      | Slide over text.
      |-
      CHSMOVE (.START_CAP_STRING [0] + 2, | # chars + size, type bytes
                START_CAP_STRING [0], | from old location
                START_CAP_STRING [2]); | to new

      !+
      | Need a negative size. The size includes the data,
      | the size byte and the type byte.
      |-
      Note that right now, the size of the normal text is the size of
      the entire capability string.

      IF .START_CAP_STRING [0] GTR 0
      THEN
          START_CAP_STRING [0] = -.START_CAP_STRING [0] - 4;
          ! size + 2 size & type bytes

          AP [PARAM_L_CUR_DATA_BYTE] = .AP [PARAM_L_CUR_DATA_BYTE] + 2;
          ! update next free byte
          SMG$DATA_OFFSET = .SMG$DATA_OFFSET + 2;
          ! update next offset
          END;                                ! first expression

      ELSE END
          ! not first byte in capability
          START_CAP_STRING [0] = .START_CAP_STRING [0] - 2;
          ! include size & type bytes for
          ! arith in count

          START_CAP_STRING [1] = SMGS$ARITH_STRING <0,8>;
          ! set (or reset) as arith string

          AP [TPASV_BLANKS] = 0;                ! ignore blanks between ( )

          RETURN (SSS_NORMAL);

      END;                                    ! end of routine INIT_ARITH_CAP

```

01FC 00000 INIT_ARITH_CAP:
 58 0000000G 00 9E 00002 .WORD Save R2,R3,R4,R5,R6,R7,R8
 MOVAB LIB\$STOP, R8

: 0877
 :

SMGSSTRING_TABL TPARSE tables for string capabilities
1-003 INIT_ARITH_CAP - Initialize arithmetic

J 14
16-Sep-1984 01:22:39
14-Sep-1984 13:10:04

VAX-11 BLiss-32 V4.0-742
[SMGRTL.SRC]SMGSTRTAB.B32:1

Page 77
(8)

57	00000000G	00	9E	00009	MOVAB	SMG\$SDATA_OFFSET, R7
52	48	53	D4	00010	CLRL	FIRST
		AC	D0	00012	MOVL	72(AP), R2
	00000000G	09	12	00016	BNEQ	1\$
68	00000000G	00	9F	00018	PUSHAB	SMG\$MISTERNAM
00		01	FB	0001E	CALLS	#1 [IB\$STOP
50	50	6C	FA	00021	CALLG	(AP), SMG\$FLUSH_SAVED_BUFFER
		AC	D0	00028	MOVL	80(AP), R0
	6240	6240	B5	0002C	TSTW	(R2)[R0]
		0A	13	0002F	BEQL	2\$
56	6240	3C	00031	MOVZWL	(R2)[R0], START_CAP_STRING	
56	48	AC	C0	00035	ADDL2	72(AP), START_CAP_STRING
		OE	11	00039	BRB	3\$
53	01	DO	0003B	MOVL	#1, FIRST	
56	4C	AC	D0	0003E	MOVL	76(AP), START_CAP_STRING
6240	67	B0	00042	MOVW	SMG\$DATA_OFFSET, -(R2)[R0]	
67	02	C0	00046	ADDL2	#2, SMG\$DATA_OFFSET	
07	53	E9	00049	BLBC	FIRST, 4\$	
AC	02	C0	0004C	ADDL2	#2, 76(AP)	
4A	53	E8	00050	BLBS	FIRST, 7\$	
50	01	A6	98	00053	CVTBL	1(START_CAP_STRING), TYPE
FFFFFE	8F	50	D1	00057	CMPL	TYPE, #2
		40	13	0005E	BEQL	8\$
		66	95	00060	TSTB	(START_CAP_STRING)
		1B	18	00062	BGEQ	5\$
7E	00000000G	00	9F	00064	PUSHAB	SMG\$SYNERR
	54	AC	7D	0006A	MOVQ	84(AP), -(SP)
	00000000G	00	DD	0006E	PUSHL	SMG\$CURRENT_LINE
		03	DD	00074	PUSHL	#3
	00000000G	00	9F	00076	PUSHAB	SMG\$ERRAT_LIN
68		06	FB	0007C	CALLS	#6 [IB\$STOP
50		66	98	0007F	CVTBL	(START_CAP_STRING), R0
50		02	C0	00082	ADDL2	#2, R0
66		50	28	00085	MOVC	R0, (START_CAP_STRING), 2(START_CAP_STRING)
		66	95	0008A	TSTB	(START_CAP_STRING)
		06	15	0008C	BLEQ	6\$
66		66	8E	0008E	MNEG	(START_CAP_STRING), (START_CAP_STRING)
66		04	82	00091	SUBB	#4, (START_CAP_STRING)
AC		02	C0	00094	ADDL2	#2, 76(AP)
67		02	C0	00098	ADDL2	#2, SMG\$DATA_OFFSET
		03	11	0009B	BRB	8\$
01	66	02	82	0009D	SUBB	#2, (START_CAP_STRING)
04	A6	02	8E	000A0	MNEG	#2, 1(START_CAP_STRING)
	AC	01	8A	000A4	BICB	#1, 4(AP)
50	01	D0	000A8	MOVL	#1, R0	
		04	000AB	RET		

; Routine Size: 172 bytes, Routine Base: _SMG\$CODE + 00AB

970 1039 1 XSBTTL 'INSERT_CARROT - Insert a ^ in the capability string'
971 1040 1 ROUTINE INSERT_CARROT =
972 1041 1
973 1042 1 ++
974 1043 1 FUNCTIONAL DESCRIPTION:
975 1044 1
976 1045 1 Stores a ^ character in the current capability string in
977 1046 1 TERMTABLE.EXE.
978 1047 1
979 1048 1 CALLING SEQUENCE:
980 1049 1 status = INSERT_CARROT ()
981 1050 1
982 1051 1 FORMAL PARAMETERS:
983 1052 1
984 1053 1
985 1054 1
986 1055 1
987 1056 1
988 1057 1
989 1058 1 AP Points to TPARSE parameter block
990 1059 1
991 1060 1 IMPLICIT OUTPUTS:
992 1061 1
993 1062 1
994 1063 1
995 1064 1 COMPLETION STATUS:
996 1065 1
997 1066 1 SSS_NORMAL
998 1067 1
999 1068 1 SIDE EFFECTS:
1000 1069 1
1001 1070 1 --
1002 1071 1
1003 1072 2 BEGIN
1004 1073 2
1005 1074 2 BUILTIN
1006 1075 2 CALLG,
1007 1076 2 AP;
1008 1077 2 MAP
1009 1078 2 AP : REF BLOCK [,BYTE];
1010 1079 2
1011 1080 2
1012 1081 2 If this is not the NAME capability and we have no pointers set up
1013 1082 2 for the terminal definition, then NAME was not the first capability
1014 1083 2 in the definition. Complain.
1015 1084 2
1016 1085 2
1017 1086 2 BEGIN
1018 1087 2 BIND
1019 1088 2 CAP_PTRS = .AP [PARAM_L_CUR_TERM_DEF] : VECTOR [,WORD];
1020 1089 2
1021 1090 2 IF CAP_PTRS EQL 0
1022 1091 2 THEN
1023 1092 2 SIGNAL_STOP (SMG\$MISTERNAM);
1024 1093 2
1025 1094 2
1026 1095 3 ! Move the capability data. The byte count is in the first byte and

```

: 1027    1096 3 | the actual data follows.
: 1028    1097 3 |
: 1029    1098 3 | Part of the string may already be copied - append in this part so
: 1030    1099 3 | as not to overwrite it. (This can happen if escape or control are
: 1031    1100 3 | part of the sequence.)
: 1032    1101 3 |
: 1033    1102 3 |
: 1034    1103 3 | CALLG (.AP, SMG$FLUSH_SAVED_BUFFER);
: 1035    1104 3 |           T copy saved string
: 1036    1105 3 |           AP [PARAM_L_SAVED_TOKENCNT] = 1;
: 1037    1106 3 |           AP [PARAM_L_SAVED_TOKENSTR] = UPLIT ('^');
: 1038    1107 3 |           ! carrot is now 'saved'
: 1039    1108 3 | CALLG (.AP, SMG$FLUSH_SAVED_BUFFER);
: 1040    1109 3 |
: 1041    1110 2 | END;                                ! end of BINDS scope
: 1042    1111 2 |
: 1043    1112 2 | RETURN (SSS_NORMAL);
: 1044    1113 2 |
: 1045    1114 1 | END;                                ! end of routine INSERT_CARROT

```

00 00 00 5E 00157 .BLKB 1
00 00 00 5E 00158 P.AAC: .ASCII \^<0><0><0>

0004 00000 INSERT_CARROT:					
52	0000000G	00	9E	00002	.WORD Save R2
48		AC	D5	00009	MOVAB SMG\$FLUSH_SAVED_BUFFER, R2
			0D	12 0000C	TSTL 72(AP)
	0000000G	00	9F	0000E	BNEQ 1\$
00		01	FB	00014	PUSHAB SMG\$ MISTERNAM
54		62	FA	0001B	CALLS #1, [IBSTOP
58	AC	01	D0	0001E	CALLG (AP), SMG\$FLUSH_SAVED_BUFFER
		D7	AF	9E 00022	MOVL #1, 84(AP)
		62	FA	00027	MOVAB P.AAC, 88(AP)
	50		01	D0 0002A	CALLG (AP), SMG\$FLUSH_SAVED_BUFFER
			04	0002D	MOVL #1, R0
					RET

: Routine Size: 46 bytes, Routine Base: _SMG\$CODE + 015C

```
1047      1115 1 %SBTTL 'INSERT_DOLLAR - Insert a $ into a capability string'
1048      1116 1 ROUTINE INSERT_DOLLAR =
1049      1117 1 ++
1050      1119 1 FUNCTIONAL DESCRIPTION:
1051      1120 1 Stores a $ character in the current capability string in
1052      1121 1 TERMTABLE.EXE.
1053      1122 1
1054      1123 1
1055      1124 1 CALLING SEQUENCE:
1056      1125 1
1057      1126 1     status = INSERT_DOLLAR ()
1058      1127 1
1059      1128 1 FORMAL PARAMETERS:
1060      1129 1
1061      1130 1     NONE
1062      1131 1
1063      1132 1 IMPLICIT INPUTS:
1064      1133 1
1065      1134 1     AP      Points to TPARSE parameter block
1066      1135 1
1067      1136 1 IMPLICIT OUTPUTS:
1068      1137 1
1069      1138 1     NONE
1070      1139 1
1071      1140 1 COMPLETION STATUS:
1072      1141 1
1073      1142 1     SSS_NORMAL
1074      1143 1
1075      1144 1 SIDE EFFECTS:
1076      1145 1
1077      1146 1
1078      1147 1
1079      1148 2
1080      1149 2
1081      1150 2
1082      1151 2
1083      1152 2
1084      1153 2
1085      1154 2
1086      1155 2
1087      1156 2
1088      1157 2
1089      1158 2
1090      1159 2
1091      1160 2
1092      1161 2
1093      1162 2
1094      1163 2
1095      1164 2
1096      1165 2
1097      1166 2
1098      1167 2
1099      1168 2
1100      1169 2
1101      1170 2
1102      1171 2
      BEGIN
      BUILTIN
      CALLG,
      AP;
      MAP
      AP : REF BLOCK [,BYTE];
      BIND
      CAP_PTRS = .AP [PARAM_L_CUR_TERM_DEF] : VECTOR [,WORD];
      If this is not the NAME capability and we have no pointers set up
      for the terminal definition, then NAME was not the first capability
      in the definition. Complain.
      IF CAP_PTRS EQL 0
      THEN
      SIGNAL_STOP (SMG$_MISTERNAM);
      Move the capability data. The byte count is in the first byte and
      the actual data follows.
```

```

: 1104 1172 2 | Part of the string may already be copied - append in this part so
: 1105 1173 2 | as not to overwrite it. (This can happen if escape or control are
: 1106 1174 2 | part of the sequence.)
: 1107 1175 2
: 1108 1176 2
: 1109 1177 2 CALLG (.AP, SMG$FLUSH_SAVED_BUFFER);
: 1110 1178 2 | copy saved string
: 1111 1179 2 AP [PARAM_L_SAVED_TOKENCNT] = 1;
: 1112 1180 2 AP [PARAM_L_SAVED_TOKENSTR] = UPLIT ('$');
: 1113 1181 2 | $ is now 'saved'
: 1114 1182 2 CALLG (.AP, SMG$FLUSH_SAVED_BUFFER);
: 1115 1183 2 | append $
: 1116 1184 2
: 1117 1185 2 RETURN (SSS_NORMAL);
: 1118 1186 2
: 1119 1187 1 END: ! end of routine INSERT_DOLLAR

```

```

00 00 00 24 0018A .BLKB 2
0018C P.AAD: .ASCII \$\<0>\<0>\<0>

```

0004 00000 INSERT_DOLLAR:					
52	00000000G	00	9E	00002	.WORD Save R2
48		AC	D5	00009	MOVAB SMG\$FLUSH_SAVED_BUFFER, R2
		0D	12	0000C	TSTL 72(AP)
00000000G	00	9F	0000E	BNEQ 1\$	
00000000G	01	FB	00014	PUSHAB SMG\$ MISTERNAM	
54		62	FA	0001B	CALLS #1, [IBSTOP
58	AC	01	D0	0001E	CALLG (AP), SMG\$FLUSH_SAVED_BUFFER
		07	AF	00022	MOVL #1, 84(AP)
		62	FA	00027	MOVAB P.AAD, 88(AP)
	50	01	D0	0002A	CALLG (AP), SMG\$FLUSH_SAVED_BUFFER
			04	0002D	MOVL #1, R0
					RET

; Routine Size: 46 bytes. Routine Base: _SMG\$CODE + 0190

```
: 1121      1188 1 ZSBTTL 'INSERT_EXCLAMATION - Insert a ! into a capability string'  
: 1122      1189 1 ROUTINE INSERT_EXCLAMATION =  
: 1123      1190 1  
: 1124      1191 1 ++  
: 1125      1192 1 FUNCTIONAL DESCRIPTION:  
: 1126      1193 1  
: 1127      1194 1 Stores a ! character in the current capability string in  
: 1128      1195 1 TERMTABLE.EXE.  
: 1129      1196 1  
: 1130      1197 1 CALLING SEQUENCE:  
: 1131      1198 1  
: 1132      1199 1 status = INSERT_EXCLAMATION ()  
: 1133      1200 1  
: 1134      1201 1 FORMAL PARAMETERS:  
: 1135      1202 1  
: 1136      1203 1 NONE  
: 1137      1204 1  
: 1138      1205 1 IMPLICIT INPUTS:  
: 1139      1206 1  
: 1140      1207 1 AP Points to TPARSE parameter block  
: 1141      1208 1  
: 1142      1209 1 IMPLICIT OUTPUTS:  
: 1143      1210 1  
: 1144      1211 1 NONE  
: 1145      1212 1  
: 1146      1213 1 COMPLETION STATUS:  
: 1147      1214 1  
: 1148      1215 1 SSS_NORMAL  
: 1149      1216 1  
: 1150      1217 1 SIDE EFFECTS:  
: 1151      1218 1  
: 1152      1219 1 --  
: 1153      1220 1  
: 1154      1221 2 BEGIN  
: 1155      1222 2 BUILTIN  
: 1156      1223 2 CALLG,  
: 1157      1224 2 AP;  
: 1158      1225 2 MAP  
: 1159      1226 2 AP : REF BLOCK [,BYTE];  
: 1160      1227 2 BIND  
: 1161      1228 2 CAP_PTRS = .AP [PARAM_L_CUR_TERM_DEF] : VECTOR [,WORD];  
: 1162      1229 2  
: 1163      1230 2 +  
: 1164      1231 2 If this is not the NAME capability and we have no pointers set up  
: 1165      1232 2 for the terminal definition, then NAME was not the first capability  
: 1166      1233 2 in the definition. Complain.  
: 1167      1234 2 -  
: 1168      1235 2  
: 1169      1236 2 IF CAP_PTRS EQL 0  
: 1170      1237 2 THEN  
: 1171      1238 2 SIGNAL_STOP (SMGS_MISTERNAM);  
: 1172      1239 2  
: 1173      1240 2 +  
: 1174      1241 2 Move the capability data. The byte count is in the first byte and  
: 1175      1242 2 the actual data follows.  
: 1176      1243 2  
: 1177      1244 2 Part of the string may already be copied - append in this part so
```

```

1178 1245 2 | as not to overwrite it. (This can happen if escape or control are
1179 1246 2 | part of the sequence.)
1180 1247 2 |
1181 1248 2 |
1182 1249 2 CALLG (.AP, SMG$FLUSH_SAVED_BUFFER);
1183 1250 2 ! copy saved buffer
1184 1251 2 AP [PARAM_L_SAVED_TOKENCNT] = 1;
1185 1252 2 AP [PARAM_L_SAVED_TOKENSTR] = UPLIT ('!!');
1186 1253 2 ! exclamation is now 'saved'
1187 1254 2 CALLG (.AP, SMG$FLUSH_SAVED_BUFFER);
1188 1255 2 ! append !
1189 1256 2 |
1190 1257 2 RETURN (SSS_NORMAL);
1191 1258 2 |
1192 1259 1 END;                                ! end of routine INSERT_EXCLAMATION

```

00 00 00 21 001BE .BLKB 2
001C0 P.AAE: .ASCII \!\\<0><0><0>

0004 00000 INSERT_EXCLAMATION:							
							.WORD Save R2
	52 00000000G	00 9E 00002			MOVAB	SMG\$FLUSH_SAVED_BUFFER, R2	: 1189
	48	AC D5 00009			TSTL	72(AP)	: 1236
	00 0000000G	0D 12 0000C			BNEQ	1\$: 1238
00000000G	00	00 9F 0000E			PUSHAB	SMG\$ MISTERNAM	: 1249
	62	01 FB 00014		1\$:	CALLS	#1, [IB\$STOP	: 1251
	54 AC	6C FA 0001B			CALLG	(AP), SMG\$FLUSH_SAVED_BUFFER	: 1252
	58 AC	01 D0 0001E			MOVL	#1, 84(AP)	: 1254
	D7	AF 9E 00022			MOVAB	P.AAE, 88(AP)	: 1257
	50	6C FA 00027			CALLG	(AP), SMG\$FLUSH_SAVED_BUFFER	: 1259
		01 D0 0002A			MOVL	#1, R0	
		04 0002D			RET		

; Routine Size: 46 bytes, Routine Base: _SMG\$CODE + 01C4

```

1194 1260 1 %SBTTL 'INSERT_PARENTHESIS - Insert a ( into a capability string'
1195 1261 1 ROUTINE INSERT_PARENTHESIS =
1196 1262 1
1197 1263 1 ++
1198 1264 1 FUNCTIONAL DESCRIPTION:
1199 1265 1
1200 1266 1 Stores a ( character in the current capability string in
1201 1267 1 TERMTABLE.EXE.
1202 1268 1
1203 1269 1 CALLING SEQUENCE:
1204 1270 1
1205 1271 1     status = INSERT_PARENTHESIS ()
1206 1272 1
1207 1273 1 FORMAL PARAMETERS:
1208 1274 1
1209 1275 1     NONE
1210 1276 1
1211 1277 1 IMPLICIT INPUTS:
1212 1278 1
1213 1279 1     AP      Points to TPARSE parameter block
1214 1280 1
1215 1281 1 IMPLICIT OUTPUTS:
1216 1282 1
1217 1283 1     NONE
1218 1284 1
1219 1285 1 COMPLETION STATUS:
1220 1286 1
1221 1287 1     SSS_NORMAL
1222 1288 1
1223 1289 1 SIDE EFFECTS:
1224 1290 1
1225 1291 1 --
1226 1292 1
1227 1293 2 BEGIN
1228 1294 2 BUILTIN
1229 1295 2     CALLG,
1230 1296 2     AP;
1231 1297 2     MAP
1232 1298 2     AP : REF BLOCK [,BYTE];
1233 1299 2     BIND
1234 1300 2     CAP_PTRS = .AP [PARAM_L_CUR_TERM_DEF] : VECTOR [,WORD];
1235 1301 2
1236 1302 2
1237 1303 2     If this is not the NAME capability and we have no pointers set up
1238 1304 2     for the terminal definition, then NAME was not the first capability
1239 1305 2     in the definition. Complain.
1240 1306 2
1241 1307 2
1242 1308 2     IF CAP_PTRS EQL 0
1243 1309 2     THEN
1244 1310 2     SIGNAL_STOP (SMGS_MISTERNAM);
1245 1311 2
1246 1312 2
1247 1313 2     Move the capability data. The byte count is in the first byte and
1248 1314 2     the actual data follows.
1249 1315 2
1250 1316 2     Part of the string may already be copied - append in this part so

```

```

: 1251      1317 2 | as not to overwrite it. (This can happen if escape or control are
: 1252      1318 2 | part of the sequence.)
: 1253      1319 2
: 1254      1320 2
: 1255      1321 2
: 1256      1322 2 | copy saved string
: 1257      1323 2 | AP [PARAM_L_SAVED_TOKENCNT] = 1;
: 1258      1324 2 | AP [PARAM_L_SAVED_TOKENSTR] = UPLIT ('(');
: 1259      1325 2 | ! ( is now saved
: 1260      1326 2
: 1261      1327 2 | append (
: 1262      1328 2
: 1263      1329 2
: 1264      1330 2
: 1265      1331 1 | end of routine INSERT_PARENTHESIS

      RETURN (SSS_NORMAL);
      END;

```

00 00 00 28 001F2 .BLKB 2
00 00 00 28 001F4 P.AAF: .ASCII \(\<0>\<0>\<0>\)

0004 00000 INSERT_PARENTHESIS:							
52	00000000G	00	9E	00002	.WORD	Save R2	1261
	48	AC	D5	00009	MOVAB	SMG\$FLUSH_SAVED_BUFFER, R2	1308
			0D	12 0000C	TSTL	72(AP)	
00000000G	00	00	9F	0000E	BNEQ	1S	
	62	01	FB	00014	PUSHAB	SMG\$ MISTERNAM	1310
54	AC	6C	FA	0001B	CALLS	#1, [IB\$STOP	1321
58	AC	01	D0	0001E	CALLG	(AP), SMG\$FLUSH_SAVED_BUFFER	1323
	62	AF	9E	00022	MOVL	#1, 84(AP)	1324
	50	D7	6C	00027	MOVAB	P.AAF, 88(AP)	1326
			01	D0 0002A	CALLG	(AP), SMG\$FLUSH_SAVED_BUFFER	1329
			04	0002D	MOVL	#1, R0	
					RET		1331

: Routine Size: 46 bytes. Routine Base: _SMG\$CODE + 01F8

```

1267 1332 1 XSBTTL 'INVALID_DIRECTIVE - signal invalid directive error'
1268 1333 1 ROUTINE INVALID_DIRECTIVE =
1269 1334 1
1270 1335 1 ++
1271 1336 1 FUNCTIONAL DESCRIPTION:
1272 1337 1
1273 1338 1 Following a !, we have found some directive we don't handle.
1274 1339 1 Signal an error.
1275 1340 1
1276 1341 1 CALLING SEQUENCE:
1277 1342 1
1278 1343 1
1279 1344 1 status = INVALID_DIRECTIVE ()
1280 1345 1
1281 1346 1 FORMAL PARAMETERS:
1282 1347 1
1283 1348 1
1284 1349 1
1285 1350 1
1286 1351 1
1287 1352 1
1288 1353 1
1289 1354 1
1290 1355 1
1291 1356 1
1292 1357 1
1293 1358 1
1294 1359 1
1295 1360 1
1296 1361 1
1297 1362 1
1298 1363 1
1299 1364 1
1300 1365 1
1301 1366 2
1302 1367 2
1303 1368 2
1304 1369 2
1305 1370 2
1306 1371 2
1307 1372 2
1308 1373 2
1309 1374 2
1310 1375 2
1311 1376 2
1312 1377 1

    BEGIN
    BUILTIN
    AP;
    MAP
    AP : REF BLOCK [,BYTE];
    SIGNAL_STOP (SMGS_ERRAT LIN,
                  3, .SMG$CURRENT_LINE,
                  .AP [TPASL-TOKENCNT],
                  .AP [TPASL-TOKENPTR],
                  SMGS_INVDIR)
END;           ! end of routine INVALID_DIRECTIVE

```

0000 00000 INVALID_DIRECTIVE:

7E	0000000G	00 9F 0002	.WORD Save nothing
	10	AC 7D 0008	PUSHAB SMGS_INVDIR
	0000000G	00 DD 0000C	MOVQ 16(AP), -(SP)
	03	DD 00012	PUSHL SMG\$CURRENT_LINE
			PUSHL #3

```

: 1333
: 1372
: 1374
: 1373
: 1372

```

SMGSSTRING_TABL TPARSE tables for string capabilities
1-003 INVALID_DIRECTIVE - signal invalid directive er 16-Sep-1984 01:22:35 VAX-11 Bliss-32 V4.0-742
6 15 14-Sep-1984 13:10:04 [SMGRTL.SRC]SMGSTRTAB.B32;1

Page 87
(13)

00000000G 00 00000000G 00 9F 00014
06 FB 0001A
04 00021

PUSHAB SMGS_ERRAT_LIN
CALLS #6, [IB\$STOP
RET

1377

; Routine Size: 34 bytes, Routine Base: _SMG\$CODE + 0226

```
1314 1378 1 %SBTTL 'STORE_OPERAND - Store arithmetic operand'  
1315 1379 1 ROUTINE STORE_OPERAND =  
1316 1380 1  
1317 1381 1 ++  
1318 1382 1 FUNCTIONAL DESCRIPTION:  
1319 1383 1  
1320 1384 1 We have just found an operand which we need to store in TERMTABLE.  
1321 1385 1 We will also check to see if an operator was previously found and  
1322 1386 1 needs to be stored.  
1323 1387 1  
1324 1388 1 CALLING SEQUENCE:  
1325 1389 1  
1326 1390 1  
1327 1391 1  
1328 1392 1  
1329 1393 1  
1330 1394 1  
1331 1395 1  
1332 1396 1  
1333 1397 1  
1334 1398 1  
1335 1399 1 AP Points to TPARSE parameter block  
1336 1400 1  
1337 1401 1  
1338 1402 1  
1339 1403 1  
1340 1404 1  
1341 1405 1  
1342 1406 1  
1343 1407 1  
1344 1408 1  
1345 1409 1  
1346 1410 1  
1347 1411 1 --  
1348 1412 1  
1349 1413 2 BEGIN  
1350 1414 2 BUILTIN  
1351 1415 2 CALLG,  
1352 1416 2 AP;  
1353 1417 2 MAP  
1354 1418 2 AP : REF BLOCK [,BYTE];  
1355 1419 2 LOCAL  
1356 1420 2 COPY_BUFFER : VECTOR [,BYTE]; ! construct bytes to copy here  
1357 1421 2  
1358 1422 2 AP [PARAM_L_SAVED_TOKENCNT] = 6; ! longword + type, size bytes  
1359 1423 2  
1360 1424 2 COPY_BUFFER [0] = SMG$K_OPERAND<0,8>; ! type  
1361 1425 2 COPY_BUFFER [1] = 0;  
1362 1426 2 CH$MOVE (4, AP [TPA$L_NUMBER], COPY_BUFFER [2]);  
1363 1427 2 ! store binary operand  
1364 1428 2  
1365 1429 2  
1366 1430 2 !+ If this is the second operand, then we need to store an operator.  
1367 1431 2  
1368 1432 2  
1369 1433 2 IF .SMG$SMASK_ADR NEQ 0  
1370 1434 2 THEN
```

```

: 1371 1435 3      BEGIN
: 1372 1436 3      COPY_BUFFER [6] = .SMG$SMASK_ADR;
: 1373 1437 3      AP [PARAM_L_SAVED_TOKENCNT] = .AP [PARAM_L_SAVED_TOKENCNT] + 1;
: 1374 1438 2      END;
: 1375 1439 2
: 1376 1440 2      AP [PARAM_L_SAVED_TOKENSTR] = COPY_BUFFER;
: 1377 1441 2
: 1378 1442 2      CALLG (.AP, SMG$FLUSH_ARITHMETIC); ! write the buffer
: 1379 1443 2
: 1380 1444 2      SMG$SMASK_ADR = 0;           ! re-init operator
: 1381 1445 2
: 1382 1446 2      RETURN (SS$_NORMAL);
: 1383 1447 2
: 1384 1448 1      END;           ! end of routine STORE_OPERAND

```

0004 00000 STORE_OPERAND:								
							.WORD	Save R2
	52	00000000G	00	9E	00002		MOVAB	SMG\$SMASK_ADR, R2
54	5E		08	C2	00009		SUBL2	#8, SP
	AC		06	D0	0000C		MOVL	#6 84(AP)
02	6E	FD	8F	9B	00010		MOVZBW	#253, COPY_BUFFER
	AE		AC	D0	00014		MOVL	28(AP), COPY_BUFFER+2
	50		62	D0	00019		MOVL	SMG\$SMASK_ADR, R0
			07	13	0001C		BEQL	1\$
06	AE		50	90	0001E		MOVB	R0, COPY_BUFFER+6
			54	AC	00022		INCL	84(AP)
58	AC	00	6E	9E	00025	1\$:	MOVAB	COPY_BUFFER, 88(AP)
			6C	FA	00029		CALLG	(AP) SMG\$FLUSH_ARITHMETIC
			62	D4	00030		CLRL	SMG\$SMASK_ADR
			50	01	00032		MOVL	#1, R0
					04		RET	

: Routine Size: 54 bytes. Routine Base: _SMG\$CODE + 0248

: 1379
: 1422
: 1424
: 1426
: 1433
: 1436
: 1437
: 1440
: 1442
: 1444
: 1446
: 1448

```

1386 1449 1 %SBTTL 'STORE_SUBSTITUTION - store substitution flag in arithmetic cap'
1387 1450 1 ROUTINE STORE_SUBSTITUTION =
1388 1451 1
1389 1452 1 !++
1390 1453 1 FUNCTIONAL DESCRIPTION:
1391 1454 1
1392 1455 1     Store an encoded flag indicating that a user argument must be
1393 1456 1     substituted into the expression at run-time.
1394 1457 1
1395 1458 1 CALLING SEQUENCE:
1396 1459 1
1397 1460 1
1398 1461 1     status = STORE_SUBSTITUTION ()
1399 1462 1
1400 1463 1 FORMAL PARAMETERS:
1401 1464 1     NONE
1402 1465 1
1403 1466 1
1404 1467 1 IMPLICIT INPUTS:
1405 1468 1     AP      Points to TPARSE parameter block
1406 1469 1
1407 1470 1 IMPLICIT OUTPUTS:
1408 1471 1     NONE
1409 1472 1
1410 1473 1
1411 1474 1
1412 1475 1 COMPLETION STATUS:
1413 1476 1     SSS_NORMAL
1414 1477 1
1415 1478 1
1416 1479 1 SIDE EFFECTS:
1417 1480 1
1418 1481 1
1419 1482 1
1420 1483 2
1421 1484 2 BEGIN
1422 1485 2     BUILTIN
1423 1486 2     CALLG,
1424 1487 2     AP;
1425 1488 2     MAP
1426 1489 2     AP : REF BLOCK [,BYTE];
1427 1490 2     LOCAL
1428 1491 2     COPY_BUFFER : VECTOR [3,BYTE];
1429 1492 2
1430 1493 2     COPY_BUFFER [0] = SMG$K SUBSTITUTE <0,8>;
1431 1494 2     COPY_BUFFER [1] = .AP [TPASL NUMBER];
1432 1495 2     ! store decimal argument number as a byte
1433 1496 2     AP [PARAM_L_SAVED_TOKENCNT] = 2;
1434 1497 2
1435 1498 2     ! See if there's a saved operator that needs to be inserted.
1436 1499 2
1437 1500 2
1438 1501 2     IF .SMG$$MASK_ADR NEQ 0
1439 1502 2     THEN
1440 1503 2     BEGIN
1441 1504 2     COPY_BUFFER [2] = .SMG$$MASK_ADR;
1442 1505 3     AP [PARAM_L_SAVED_TOKENCNT] = .AP [PARAM_L_SAVED_TOKENCNT] + 1;

```

```

1443 1506 3      SMG$SMASK_ADR = 0;
1444 1507 2      END;
1445 1508 2
1446 1509 2      AP [PARAM_L_SAVED_TOKENSTR] = COPY_BUFFER;
1447 1510 2
1448 1511 2      CALLG (.AP, SMG$FLUSH_ARITHMETIC);
1449 1512 2          ! move buffer to data area
1450 1513 2
1451 1514 2      RETURN (SS$_NORMAL);
1452 1515 2
1453 1516 1      END;                                ! end of routine STORE_SUBSTITUTION

```

0004 00000 STORE_SUBSTITUTION:					
				.WORD	Save R2
	52	00000000G	00	9E 00002	MOVAB SMG\$SMASK_ADR, R2
	5E		04	C2 00009	SUBL2 #4, SP
01	6E		04	8E 0000C	MNEG B #4, COPY_BUFFER
	AE	1C	AC	90 0000F	MOVB 28(AP), COPY_BUFFER+1
54	AC		02	D0 00014	MOVL #2, 84(AP)
	50		62	D0 00018	MOVL SMG\$SMASK_ADR, R0
02	AE		09	13 0001B	BEQL 1\$
	54	54	50	90 0001D	MOVB R0, COPY_BUFFER+2
	58	AC	AC	D6 00021	INCL 84(AP)
00000000G	00		62	D4 00024	CLRL SMG\$SMASK_ADR
	50		6E	9E 00026	MOVAB COPY_BUFFER, 88(AP)
			6C	FA 0002A	CALLG (AP), SMG\$FLUSH_ARITHMETIC
			01	D0 00031	MOVL #1, R0
			04	00034	RET

; Routine Size: 53 bytes, Routine Base: _SMG\$CODE + 027E

```

1455      1517 1 %SBTTL 'NOT_STRING - signal an unknown capability name'
1456      1518 1 ROUTINE NOT_STRING =
1457      1519 1
1458      1520 1 ++
1459      1521 1 FUNCTIONAL DESCRIPTION:
1460      1522 1
1461      1523 1      We just found a unknown capability name. It could be a misspelling
1462      1524 1      or it could be a name misplaced under the wrong heading. Signal an
1463      1525 1      error.
1464      1526 1
1465      1527 1 CALLING SEQUENCE:
1466      1528 1
1467      1529 1
1468      1530 1      status = NOT_STRING ()
1469      1531 1
1470      1532 1 FORMAL PARAMETERS:
1471      1533 1      NONE
1472      1534 1
1473      1535 1
1474      1536 1 IMPLICIT INPUTS:
1475      1537 1
1476      1538 1      AP      Points to TPARSE parameter block
1477      1539 1
1478      1540 1 IMPLICIT OUTPUTS:
1479      1541 1
1480      1542 1      NONE
1481      1543 1
1482      1544 1
1483      1545 1
1484      1546 1      SSS_NORMAL
1485      1547 1
1486      1548 1
1487      1549 1 SIDE EFFECTS:
1488      1550 1      --
1489      1551 1
1490      1552 2      BEGIN
1491      1553 2      BUILTIN
1492      1554 2      AP;
1493      1555 2      MAP
1494      1556 2      AP : REF BLOCK [,BYTE];
1495      1557 2
1496      1558 2      SIGNAL_STOP (SMGS$ERRAT LIN,
1497      1559 2      3, .SMG$SCURRENT_LINE,
1498      1560 2      .AP [TPASL_TOKENCNT],
1499      1561 2      .AP [TPASL_TOKENPTR],
1500      1562 2      SMGS_NOTSTRCAP)
1501      1563 1      END:      ! end of routine NOT_STRING

```

0000 0000 NOT_STRING:

7E	0000000G	00	9F	00002	.WORD	Save nothing	: 1518
	10	AC	7D	00008	PUSHAB	SMGS_NOTSTRCAP	: 1558
	0000000G	00	DD	0000C	MOVQ	16(AP), -(SP)	: 1560
					PUSHL	SMG\$SCURRENT_LINE	: 1559

SMG\$STRING_TABL TPARSE tables for string capabilities M 15
1-003 NOT_STRING - signal an unknown capability name 16-Sep-1984 01:22:35 VAX-11 Bliss-32 V4.0-742
[SMGRTL.SRC]SMG\$TRTAB.B32;1 Page 93
(16)

00000000G 00 00000000G 03 DD 00012 PUSHL #3
00000000G 00 00000000G 00 9F 00014 PUSHAB SMG\$ERRAT LIN
00000000G 00 00000000G 06 FB 0001A CALLS #6, [IBSSTOP
00000000G 00 00000000G 04 00021 RET

; Routine Size: 34 bytes, Routine Base: _SMG\$CODE + 0283

; 1502 1564 1 !<BLF/PAGE>

SMG\$STRING_TABL TPARSE tables for string capabilities
1-003 NOT_STRING - signal an unknown capability name

N 15
16-Sep-1984 01:22:35 VAX-11 Bliss-32 v4.0-742
14-Sep-1984 13:10:04 [SMGRTL.SRC]SMG\$RTAB.B32:1

Page 94
(17)

: 1504 1565 1 END
: 1505 1566 1
: 1506 1567 0 ELUDOM

: ! End of module SMG\$STRING_TABLES

.EXTRN LIB\$STOP

PSECT SUMMARY

Name	Bytes	Attributes								
_LIB\$KEY0\$	342	NOVEC,NOWRT,	RD	:	EXE,	SHR,	LCL,	REL,	CON,	PIC,ALIGN(1)
_LIB\$STATES	2576	NOVEC,NOWRT,	RD	:	EXE,	SHR,	LCL,	REL,	CON,	PIC,ALIGN(1)
_LIB\$KEY1\$	2061	NOVEC,NOWRT,	RD	:	EXE,	SHR,	LCL,	REL,	CON,	PIC,ALIGN(1)
_SMG\$CODE	725	NOVEC,NOWRT,	RD	:	EXE,	SHR,	LCL,	REL,	CON,	PIC,ALIGN(2)

Library Statistics

File	-----	Symbols	-----	Pages	Processing
	Total	Loaded	Percent	Mapped	Time
-\$255\$DUA28:[SYSLIB]STARLET.L32:1	9776	172	1	581	00:01.0
-\$255\$DUA28:[SMGRTL.OBJ]RTLLIB.L32:1	36	0	0	8	00:00.1
-\$255\$DUA28:[SMGRTL.OBJ]SMGLIB.L32:1	469	0	0	38	00:00.4
-\$255\$DUA28:[SMGRTL.OBJ]SMGTPALIB.L32:1	41	14	34	10	00:00.1
-\$255\$DUA28:[SYSLIB]TPAMAC.L32:1	42	30	71	14	00:00.1

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LISS:SMG\$RTAB/OBJ=OBJ\$:SMG\$RTAB MSRC\$:SMG\$RTAB/UPDATE=(ENH\$:SMG\$RTAB)

Size: 694 code + 5010 data bytes
Run Time: 03:48.4
Elapsed Time: 10:51.1
Lines/CPU Min: 411
Lexemes/CPU-Min:113340
Memory Used: 758 pages
Compilation Complete

0361 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SMGTPALIB
LIS

SMGSTATAB
LIS

SMGSTRTAB
LIS